PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-211151

(43)Date of publication of application : 03.08.2001

---

(51)Int.Cl. H04L 9/08
G09C 1/00
G11B 20/10
H04L 9/32

---

(21)Application number : 2000-016213 (71)Applicant : SONY CORP

(22)Date of filing : 25.01.2000 (72)Inventor : ISHIBASHI YOSHITO
ASANO TOMOYUKI
AKISHITA TORU
SHIRAI TAIZO

---

(54) DEVICE AND METHOD FOR DATA PROCESSING CONTENTS DATA VERIFICATION VALUE IMPARTING METHOD, AND PROGRAM PROVIDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a data processor which efficiently confirms validity of the data.

SOLUTION: Contents check values are generated with contents block data of objects of verification included in data as the unit and the generated contents check value is collated to verify the validity of the contents block data units in the data. The contents

check values are generated by generating contents intermediate values by a ciphering process based upon the contents block data to be verified, e.g. DES-CBC mode deciphering and performing a ciphering process applied with a contents check value generating key for the generated contents intermediate values. A verifying process corresponding to the use style of the contents data can be performed to enable efficient verification.

LEGAL STATUS [Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] The cipher-processing section which is the data processor which processes the contents data offered by a storage or communication media, and performs cipher processing to said contents data, It has the control section which performs control to said cipher-processing section. Said cipher-processing section Per contents block data for [which is contained in data] verification, generate a contents check value and by performing collating processing of the generated contents check value The data processor characterized by having the configuration which performs justification verification processing of the contents block data unit in said data.
[Claim 2] It is the data processor according to claim 1 characterized by being the configuration which said data processor has a contents check value generation key, and said cipher-processing section generates a contents mean value based on the contents block data for verification, performs cipher processing which applied said contents check value generation key to this contents mean value, and generates a contents check value.
[Claim 3] Said cipher-processing section is a data processor according to claim 2 characterized by being the configuration which performs data processing predetermined in a predetermined cutting tool unit for the whole contents block data, and generates a contents mean value when data processing predetermined in a predetermined cutting tool unit is performed for the whole decode sentence obtained by decode processing of this contents block data when the contents block data for verification is enciphered, a contents mean value is generated and the contents block data for verification is not enciphered.
[Claim 4] Said predetermined data processing applied in generation processing of said middle check value in said cipher-processing section is a data processor according to claim 3 characterized by being EXCLUSIVE OR operation.
[Claim 5] For said decode processing applied to contents mean value generation processing in case it has a cipher-processing configuration by the CBC mode and the contents block data for verification is enciphered, said cipher-processing section is a data processor according to claim 3 characterized by being decode processing by the

CBC mode.

[Claim 6] The cipher-processing configuration by the CBC mode which said cipher-processing section has is a data processor according to claim 5 characterized by being the configuration that the count common key encryptosystem processing of double is applied only in a part of message train used as a processing object.

[Claim 7] Said cipher-processing section be the data processor according to claim 1 characterize by to have the configuration which perform justification verification processing for every contents block data unit in said data by perform collating processing of the contents check value which generated and generated the contents check value based on the parts for verification when some parts which two or more parts be contain in contents block data , and be contain in this contents block data be the candidates for verification .

[Claim 8] [when the number of the important point verification parts said whose cipher-processing sections two or more parts are contained in said contents block data, and are the candidates for verification is one] When said important point verification parts are enciphered, the whole decode sentence obtained by decode processing of important point verification parts to the value which carried out exclusive OR per predetermined cutting tool Perform cipher processing which applied the contents check value generation key, and a contents check value is generated. When said important point verification parts are not enciphered, the value which carried out exclusive OR of these whole important point verification parts per predetermined cutting tool The data processor according to claim 7 characterized by being the configuration which performs cipher processing with the application of said contents check value generation key, and generates a contents check value.

[Claim 9] [when the important point verification parts said whose cipher-processing section two or more parts are contained in said contents block data, and is a candidate for verification are plurality] As opposed to the connection data of the parts check value acquired by performing cipher processing with the application of the contents check value generation key for every parts The data processor according to claim 7 characterized by being the configuration which makes the result obtained by performing cipher processing which furthermore applied said contents check value generation key a contents check value.

[Claim 10] Said data processor is a data processor according to claim 1 characterized by having the storage device which stores the contents data which contain further the contents block data with which justification verification was performed in said cipher-processing section.

[Claim 11] It is the data processor according to claim 10 characterized by said control section having the configuration which stops the storing processing to said storage device when collating is not materialized in collating processing of the contents check value in said cipher-processing section.

[Claim 12] Said data processor is a data processor according to claim 1 characterized by having the regeneration section which reproduces further the data with which justification verification was performed in said cipher-processing section.

[Claim 13] It is the data processor according to claim 12 characterized by said control section having the configuration which stops regeneration in said regeneration section when collating is not materialized in collating processing of a contents check value [in / in said data processor / said cipher-processing section].

[Claim 14] The data-processing approach which generates a contents check value and is characterized by performing justification verification processing of the contents block data unit in said data by performing collating processing of the generated contents check value per contents block data for [which is the data-processing approach of processing the contents data offered by a storage or communication media, and is contained in data] verification.

[Claim 15] Said data-processing approach is the data-processing approach according to claim 14 characterized by performing cipher processing which applied the contents check value generation key to the contents mean value which generated and generated the contents mean value based on the contents block data for verification, and generating a contents check value.

[Claim 16] The data-processing approach according to claim 14 characterized by performing data processing predetermined in a predetermined cutting tool unit for the whole contents block data, and generating a contents mean value when data processing predetermined in a predetermined cutting tool unit is performed for the whole decode sentence obtained by decode processing of this contents block data in said data-processing approach when the contents block data for verification is enciphered, a contents mean value is generated and the contents block data for verification is not enciphered.

[Claim 17] Said predetermined data processing applied in generation processing of said middle check value in said data-processing approach is the data-processing approach according to claim 16 characterized by being EXCLUSIVE OR operation.

[Claim 18] Said decode processing applied to contents mean value generation processing in case the contents block data for verification is enciphered in generation processing of said contents mean value is the data-processing approach according to

claim 16 characterized by being decode processing by the CBC mode.
[Claim 19] The decode processing configuration by said CBC mode is the data-processing approach according to claim 18 characterized by the count common key encryptosystem processing of double applying only in a part of message train used as a processing object.
[Claim 20] The data-processing approach according to claim 14 characterized by to perform justification verification processing for every contents block data unit in said data by performing collating processing of the contents check value which generated and generated the contents check value based on the parts for verification when some parts which two or more parts are contained in contents block data, and are contained in this contents block data in said data-processing approach are the candidates for verification.
[Claim 21] Two or more parts are contained in contents block data in said data-processing approach. When the number of the important point verification parts which are the candidates for verification is one and said important point verification parts are enciphered, To the value which carried out exclusive OR per predetermined cutting tool, the whole decode sentence obtained by decode processing of important point verification parts Perform cipher processing which applied the contents check value generation key, and a contents check value is generated. The data-processing approach according to claim 20 characterized by performing cipher processing for the value which carried out exclusive OR of these whole important point verification parts per predetermined cutting tool with the application of said contents check value generation key, and generating a contents check value when said important point verification parts are not enciphered.
[Claim 22] Two or more parts are contained in said contents block data in said data-processing approach. As opposed to the connection data of the parts check value acquired by performing cipher processing with the application of the contents check value generation key for every parts when the important point verification parts which are the candidates for verification were plurality The data-processing approach according to claim 20 characterized by making into a contents check value the result obtained by performing cipher processing which furthermore applied said contents check value generation key.
[Claim 23] Said data-processing approach is the data-processing approach according to claim 14 characterized by including the step which stores the contents data which contain further the contents block data with which justification verification was performed.

[Claim 24] It is the data-processing approach according to claim 23 characterized by said control section stopping the storing processing to said storage device when, as for said data-processing approach, collating is not further materialized in collating processing of a contents check value.
[Claim 25] Said data-processing approach is the data-processing approach according to claim 14 characterized by including the step which performs regeneration which reproduces further the data with which justification verification was performed.
[Claim 26] Said data-processing approach is the data-processing approach according to claim 25 characterized by stopping regeneration in collating processing of a contents check value when collating is not materialized.
[Claim 27] The contents data verification value grant approach which generates a contents check value and is characterized by giving the generated contents check value to the contents data containing the contents block data for verification per contents block data for [which is the contents data verification value grant approach for contents data verification processing, and is contained in data] verification.
[Claim 28] Said contents check value is the contents data verification value grant approach according to claim 27 which makes a message the contents block data used as the candidate for a check, and is characterized by being the value generated by cipher processing which applied the contents check value generation key.
[Claim 29] Said contents check value is the contents data verification value grant approach according to claim 27 characterized by being the value which generates a contents mean value based on the contents block data for verification, and is generated by performing cipher processing which applied said contents check value generation key to this contents mean value.
[Claim 30] Said contents check value is the contents data verification value grant approach according to claim 27 characterized by being the value generated by performing cipher processing by the CBC mode to the contents block data for verification.
[Claim 31] The cipher-processing configuration by said CBC mode is the contents data verification value grant approach according to claim 30 characterized by being the configuration that the count common key encryptosystem processing of double is applied only in a part of message train used as a processing object.
[Claim 32] The contents data verification value grant approach according to claim 27 characterized by giving the contents check value which generated and generated the contents check value based on the parts for verification when making applicable to verification some parts which two or more parts are contained in contents block data,

and are contained in this contents block data in said contents data verification value grant approach to the contents data containing the contents block data for verification.

[Claim 33] [when the number of the important point verification parts which two or more parts are contained in said contents block data, and are the candidates for verification is one in said contents data verification value grant approach] When said important point verification parts are enciphered, the whole decode sentence obtained by decode processing of important point verification parts to the value which carried out exclusive OR per predetermined cutting tool Perform cipher processing which applied the contents check value generation key, and a contents check value is generated. When said important point verification parts are not enciphered, the value which carried out exclusive OR of these whole important point verification parts per predetermined cutting tool With the application of said contents check value generation key, perform cipher processing, and a contents check value is generated. The contents data verification value grant approach according to claim 32 characterized by giving the generated contents check value to the contents data containing the contents block data for verification.

[Claim 34] [when the important point verification parts which two or more parts are contained in said contents block data, and are the candidates for verification are plurality in said contents data verification value grant approach] As opposed to the connection data of the parts check value acquired by performing cipher processing with the application of the contents check value generation key for every parts The result obtained by performing cipher processing which furthermore applied said contents check value generation key is made into a contents check value. The contents data verification value grant approach according to claim 32 characterized by giving the generated contents check value to the contents data containing the contents block data for verification.

[Claim 35] it be the program offer medium be the program offer medium which offer the computer program which make data processing which process the contents data offer by a storage or communication media perform on computer system , and carry out that said computer program contain the step which generate a contents check value per contents block data for [which be contain in data] verification , and the step which perform justification verification processing of the contents block data unit in said data by perform collating processing of the contents check value which generated as the description .

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the grant approach of the approach of verifying the justification of the data which constitute data contents in a detail further about a program offer medium at a data processor, the data-processing approach and the contents data verification value grant approach, and a list, i.e., the existence of an alteration, equipment, and a verification value.

[0002] This invention Storages, such as DVD and CD, or CATV, the Internet, Voice available in paths, such as cables, such as satellite communication, and wireless each means of communications, an image, While reproducing in the record regenerator which a user owns and storing various contents, such as a game and a program, in the storage device of dedication, for example, a memory card, a hard disk, CD-R, etc. In case the contents stored in the storage device are used, while realizing the configuration which attaches use restricting [which a contents distribution side wishes], it is related with the configuration and approach of securing security so that unjust use of these distributed contents may not be carried out at any third persons other than a registered user.

[0003]

[Description of the Prior Art] Various software data (these are hereafter called contents (Content)), such as a game program, voice data, image data, and a document

preparation program, are circulating through networks, such as the Internet, these days through the storage of DVD, CD, etc. which can be circulated. These circulation contents can be stored in the storage device attached to record playback devices which a user owns, such as PC (Personal Computer) and a game device, for example, a memory card, a hard disk, etc., and once it is stored, they become available by playback from a storing medium.

[0004] The main components of the memory card equipment used in information machines and equipment, such as the conventional video game device and PC, are the control means for motion control, a connector for connecting with the slot which was connected to the control means and prepared in the body of information machines and equipment, the nonvolatile memory for connecting with a control means and memorizing data, etc. The nonvolatile memory with which the memory card was equipped is constituted by EEPROM, the flash memory, etc.

[0005] Various contents, such as data memorized by such memory card or a program, are called from nonvolatile memory by directions of the user through the input means connected [which were connected and was user-directed] from bodies of information machines and equipment used as a playback device, such as a game device and PC, and are reproduced through the body of information machines and equipment or the connected display, a loudspeaker, etc.

[0006] Generally as for many software contents, such as a game program, music data, and image data, the right of distribution etc. is held by the implementer and the vender. Therefore, it is common to permit use of software, and for reproduction without authorization etc. to be made not to be performed, namely, to take the configuration in consideration of security only to a fixed use limit, i.e., a regular user, on the occasion of distribution of these contents.

[0007] One technique of realizing the use limit to a user is encryption processing of distribution contents. That is, while distributing various contents, such as voice data enciphered, for example through the Internet etc., image data, and a game program, it is a means, i.e., the configuration which gives a decode key, to decode the distributed encryption contents only to those who were checked as he is a registered user.

[0008] Encryption data can be returned to available decode data (plaintext) by decryption processing in a predetermined procedure. The data encryption and the decryption approach of using an encryption key for encryption processing of such information, and using a decryption key for decryption processing are well learned from the former.

[0009] Although it is seeds, there are various methods currently called the so-called

common key encryptosystem-ized method as the one example in the mode of the data encryption and the decryption approach using an encryption key and a decryption key. A common key encryptosystem-ized method gives the encryption key used for data encryption processing, and the common key which uses for these encryption processing and a decryption the decryption key used for a decryption of data as a common thing at the user of normal, and eliminates the data access by the inaccurate user without a key. DES (data code criterion: Deta encryption standard) is in the typical method of this method.

[0010] On the other hand, for example based on a certain password etc., a Hash Function etc. can obtain the encryption key and decryption key which are used for above-mentioned encryption processing and a decryption with the application of a tropism function. On the other hand with a tropism function, the function which becomes very difficult asks for an input conversely from the output. For example, on the other hand, a tropism function is applied by considering the password which the user decided as an input, and an encryption key and a decryption key are generated based on the output. Thus, the parenchyma top of asking for the password which is the original data conversely from the obtained encryption key and a decryption key becomes impossible.

[0011] Moreover, the method which made a different algorithm processing with the encryption key used when enciphering, and processing of the decryption key used when decoding is a method called the so-called public-key-encryption-ized method. An unspecified user is the approach of using an usable public key, and a public-key-encryption-ized method performs encryption processing using the public key with which the specific individual published the encryption document to a specific individual. The decode processing of the document enciphered with the public key is attained only with the private key corresponding to the public key used for the encryption processing. Since only the individual who published the public key owns a private key, only an individual with a private key can decode the document enciphered with the public key. A RSA (Rivest-Shamir-Adleman) code is one of the typical things of a public-key-encryption-ized method.

[0012] By using such a cipher system, the system which enables the decode of encryption contents only to a registered user becomes possible. The conventional contents distribution configuration which adopted these cipher systems is briefly explained using drawing 1 .

[0013] In the playback means 10, such as PC (personal computer) and a game device, drawing 1 shows the example of a configuration which enabled the storage of the data

acquired from DVD, CD30, and Internet 40 grade for the storage means 20, such as a floppy disk, a memory card, and a hard disk, while reproducing the program acquired from the data offer means of DVD, CD30, and Internet 40 grade, voice data, image data, etc. (contents (Content)).

[0014] Encryption processing is made and the user who has the playback means 10 is provided with contents, such as a program, voice data, and image data. A registered user acquires the key data which are the encryption and a decryption key with encryption data.

[0015] The playback means 10 has CPU12 and performs regeneration of input data in the regeneration section 14. The regeneration section 14 performs decode processing of encryption data, and performs contents playbacks, such as playback of the offered program, voice data, and image data.

[0016] In order to use the offered program again, as for a registered user, a program/data carries out preservation processing of contents to the storage means 20. For the playback means 10, it has the preservation processing section 13 for performing this contents preservation processing. In order that the preservation processing section 13 may prevent the unauthorized use of the data memorized by the storage means 20, it performs encryption processing to data and performs preservation processing.

[0017] In case contents are enciphered, the key for contents codes is used. Using the key for contents codes, the preservation processing section 13 enciphers contents and memorizes it in the storage section 21 of the storage means 20, such as FD (floppy disk), a memory card, and a hard disk.

[0018] When taking out storing contents from the storage means 20 and reproducing, from the storage means 20, in the regeneration section 14 of the playback means 10, encryption data are taken out, and a user performs decode processing using the key, i.e., the decryption key, for contents decode, he acquires decode data from encryption data, and is reincarnated.

[0019] If the conventional example of a configuration shown in drawing 1 is followed, since storing contents are enciphered, with the storage means 20, such as a floppy disk and a memory card, prevention of unjust read-out from the outside will be attained. However, it will become unreproducible if it is going to play and use this floppy disk with the playback means of information machines and equipment, such as other PCs and a game device, and it is not a playback means to have the same decryption key for decoding the same contents key, i.e., the enciphered contents. Therefore, in order to realize an available gestalt in two or more information machines

and equipment, it is necessary to communalize the cryptographic key with which a user is provided.

[0020] However, communalizing the cryptographic key of contents will raise possibility of circulating the key for cipher processing disorderly to a user without a normal license, it has a fault of it becoming impossible to prevent unjust use of the contents by the user without the license of normal, and exclusion of the unjust use in PC, a game device, etc. without a normal license becomes difficult.

[0021] Furthermore, it sets by the environment which communalized the key as mentioned above. For example, the enciphered contents which were created on a certain PC and saved for storage means, such as a memory card and a floppy disk The use gestalt using not original contents data but a duplicate floppy disk becomes reproducing easily in another floppy disk is possible, and possible. In information machines and equipment, such as a game device and PC, much available contents data may have been reproduced or it may have been altered.

[0022]

[Problem(s) to be Solved by the Invention] The method of performing data verification is performed from the former by including the check value for verification in contents data, in order to confirm that the justification of contents data, i.e., data, is not altered, and carrying out collating processing of the check value generated based on the data for verification, and the check value included in contents data in a record regenerator.

[0023] However, as for the check value over data contents, being generated to the whole data is common, and in order to perform collating processing of the check value generated to the whole data, it is necessary to perform check value generation processing to the whole data used as the candidate for a check. For example, when the message authenticator (MAC) generated in DES-CBC mode performs the technique of calculating the check value ICV, it is necessary to perform processing of DES-CBC to the whole data. This computational complexity will increase as a data length becomes long, and it has a problem in respect of processing effectiveness.

[0024] It aims at providing with a program offer medium the data processor which makes it possible for this invention to solve the trouble of such a conventional technique, to perform check processing of data justification efficiently, and to perform efficiently increase in efficiency of verification processing of contents data, download processing of further as opposed to the storage device after verification, or regeneration after verification, the data-processing approach and the data verification value grant approach, and a list.

[0025]

[Means for Solving the Problem] The cipher-processing section which the 1st side face of this invention is a data processor which processes the contents data offered by a storage or communication media, and performs cipher processing to said contents data, It has the control section which performs control to said cipher-processing section. Said cipher-processing section Per contents block data for [which is contained in data] verification, generate a contents check value and by performing collating processing of the generated contents check value It is in the data processor characterized by having the configuration which performs justification verification processing of the contents block data unit in said data.

[0026] Furthermore, it is characterized by being the configuration which the data processor of this invention sets like 1 operative condition, and said data processor has a contents check value generation key, and said cipher-processing section generates a contents mean value based on the contents block data for verification, performs cipher processing which applied said contents check value generation key to this contents mean value, and generates a contents check value.

[0027] Furthermore, it is characterized by to be the configuration which the data processor of this invention sets like 1 operative condition, performs data processing predetermined in a predetermined cutting tool unit for the whole contents block data when said cipher-processing section performs data processing predetermined in a predetermined cutting tool unit for the whole decode sentence obtained by decode processing of this contents block data when the contents block data for verification is enciphered, and generates a contents mean value and the contents block data for verification is not enciphered, and generates a contents mean value.

[0028] Furthermore, the data processor of this invention sets like 1 operative condition, and it is characterized by said predetermined data processing applied in generation processing of said middle check value in said cipher-processing section being EXCLUSIVE OR operation.

[0029] Furthermore, the data processor of this invention sets like 1 operative condition, said cipher-processing section has a cipher-processing configuration by the CBC mode, and said decode processing applied to contents mean value generation processing in case the contents block data for verification is enciphered is characterized by being decode processing by the CBC mode.

[0030] Furthermore, the data processor of this invention sets like 1 operative condition, and the cipher-processing configuration by the CBC mode which said cipher-processing section has is characterized by being the configuration that the count common key encryptosystem processing of double is applied only in a part of

message train used as a processing object.

[0031] The data processor of this invention sets like 1 operative condition. Furthermore, said cipher-processing section [when some parts which two or more parts are contained in contents block data, and are contained in this contents block data are the candidates for verification] It is characterized by having the configuration which performs justification verification processing for every contents block data unit in said data by performing collating processing of the contents check value which generated and generated the contents check value based on the parts for verification.

[0032] The data processor of this invention sets like 1 operative condition. Furthermore, said cipher-processing section [when the number of the important point verification parts which two or more parts are contained in said contents block data, and are the candidates for verification is one] When said important point verification parts are enciphered, the whole decode sentence obtained by decode processing of important point verification parts to the value which carried out exclusive OR per predetermined cutting tool Perform cipher processing which applied the contents check value generation key, and a contents check value is generated. When said important point verification parts are not enciphered, it is characterized by being the configuration which performs cipher processing for the value which carried out exclusive OR of these whole important point verification parts per predetermined cutting tool with the application of said contents check value generation key, and generates a contents check value.

[0033] The data processor of this invention sets like 1 operative condition. Furthermore, said cipher-processing section [when the important point verification parts which two or more parts are contained in said contents block data, and are the candidates for verification are plurality] As opposed to the connection data of the parts check value acquired by performing cipher processing with the application of the contents check value generation key for every parts It is characterized by being the configuration which makes the result obtained by performing cipher processing which furthermore applied said contents check value generation key a contents check value.

[0034] Furthermore, the data processor of this invention sets like 1 operative condition, and said data processor is characterized by having the storage device which stores the contents data which contain further the contents block data with which justification verification was performed in said cipher-processing section.

[0035] Furthermore, when the data processor of this invention sets like 1 operative condition and collating is not materialized in collating processing of the contents

check value in said cipher-processing section, said control section is characterized by having the configuration which stops the storing processing to said storage device.

[0036] Furthermore, the data processor of this invention sets like 1 operative condition, and said data processor is characterized by having the regeneration section which reproduces further the data with which justification verification was performed in said cipher-processing section.

[0037] Furthermore, when the data processor of this invention sets like 1 operative condition and collating is not materialized in collating processing of a contents check value [in / in said data processor / said cipher-processing section], said control section is characterized by having the configuration which stops regeneration in said regeneration section.

[0038] Furthermore, the 2nd side face of this invention is the data-processing approach of processing the contents data offered by a storage or communication media, generates a contents check value and is by performing collating processing of the generated contents check value per contents block data for [which is contained in data] verification in the data-processing approach characterized by to perform justification verification processing of the contents block data unit in said data.

[0039] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and said data-processing approach is characterized by performing cipher processing which applied the contents check value generation key to the contents mean value which generated and generated the contents mean value based on the contents block data for verification, and generating a contents check value.

[0040] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and it sets to said data-processing approach. When the contents block data for verification is enciphered, perform data processing predetermined in a predetermined cutting tool unit for the whole decode sentence obtained by decode processing of this contents block data, and a contents mean value is generated. When the contents block data for verification is not enciphered, it is characterized by performing data processing predetermined in a predetermined cutting tool unit for the whole contents block data, and generating a contents mean value.

[0041] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and it is characterized by said predetermined data processing applied in generation processing of said middle check value being EXCLUSIVE OR operation in said data-processing approach.

[0042] Furthermore, the data-processing approach of this invention sets like 1

operative condition, and said decode processing applied to contents mean value generation processing in case the contents block data for verification is enciphered is characterized by being decode processing by the CBC mode in generation processing of said contents mean value.

[0043] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and the decode processing configuration by said CBC mode is characterized by the count common key encryptosystem processing of double applying only in a part of message train used as a processing object.

[0044] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and it sets to said data-processing approach. When some parts which two or more parts are contained in contents block data, and are contained in this contents block data are the candidates for verification, It is characterized by performing justification verification processing for every contents block data unit in said data by performing collating processing of the contents check value which generated and generated the contents check value based on the parts for verification.

[0045] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and it sets to said data-processing approach. When the number of the important point verification parts which two or more parts are contained in contents block data, and are the candidates for verification is one, When said important point verification parts are enciphered, the whole decode sentence obtained by decode processing of important point verification parts to the value which carried out exclusive OR per predetermined cutting tool Perform cipher processing which applied the contents check value generation key, and a contents check value is generated. When said important point verification parts are not enciphered, it is characterized by performing cipher processing for the value which carried out exclusive OR of these whole important point verification parts per predetermined cutting tool with the application of said contents check value generation key, and generating a contents check value.

[0046] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and it sets to said data-processing approach. When the important point verification parts which two or more parts are contained in said contents block data, and are the candidates for verification are plurality, It is characterized by making into a contents check value the result obtained by performing cipher processing which applied said contents check value generation key further to the connection data of the parts check value acquired by performing cipher processing with the application of the contents check value generation key for every parts.

[0047] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and said data-processing approach is characterized by including the step which stores the contents data which contain further the contents block data with which justification verification was performed.

[0048] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and when, as for said data-processing approach, collating is not further materialized in collating processing of a contents check value, said control section is characterized by stopping the storing processing to said storage device.

[0049] Furthermore, the data-processing approach of this invention sets like 1 operative condition, and said data-processing approach is characterized by including the step which performs regeneration which reproduces further the data with which justification verification was performed.

[0050] Furthermore, it is characterized by for the data-processing approach of this invention setting like 1 operative condition, and said data-processing approach stopping regeneration in collating processing of a contents check value, when collating is not materialized.

[0051] Furthermore, the 3rd side face of this invention is the contents data verification value grant approach for contents data verification processing, per contents block data for [which is contained in data] verification, generates a contents check value and is in the contents data verification value grant approach characterized by giving the generated contents check value to the contents data containing the contents block data for verification.

[0052] Furthermore, the contents data verification value grant approach of this invention sets like 1 operative condition, said contents check value makes a message the contents block data used as the candidate for a check, and it is characterized by being the value generated by cipher processing which applied the contents check value generation key.

[0053] Furthermore, it is characterized by being the value which the contents data verification value grant approach of this invention sets like 1 operative condition, and said contents check value generates a contents mean value based on the contents block data for verification, and is generated by performing cipher processing which applied said contents check value generation key to this contents mean value.

[0054] Furthermore, the contents data verification value grant approach of this invention sets like 1 operative condition, and said contents check value is characterized by being the value generated by performing cipher processing by the CBC mode to the contents block data for verification.

[0055] Furthermore, the contents data verification value grant approach of this invention sets like 1 operative condition, and the cipher-processing configuration by said CBC mode is characterized by being the configuration that the count common key encryptosystem processing of double is applied only in a part of message train used as a processing object.

[0056] Furthermore, when making applicable to verification some parts which the contents data verification value grant approach of this invention sets like 1 operative condition, and two or more parts are contained in contents block data, and are contained in this contents block data, it is characterized by giving the contents check value which generated and generated the contents check value based on the parts for verification to the contents data containing the contents block data for verification.

[0057] Furthermore, the contents data verification value grant approach of this invention sets like 1 operative condition. [when the number of the important point verification parts which two or more parts are contained in said contents block data, and are the candidates for verification is one] When said important point verification parts are enciphered, the whole decode sentence obtained by decode processing of important point verification parts to the value which carried out exclusive OR per predetermined cutting tool Perform cipher processing which applied the contents check value generation key, and a contents check value is generated. When said important point verification parts are not enciphered, the value which carried out exclusive OR of these whole important point verification parts per predetermined cutting tool It is characterized by performing cipher processing with the application of said contents check value generation key, and giving the contents check value which generated and generated the contents check value to the contents data containing the contents block data for verification.

[0058] Furthermore, the contents data verification value grant approach of this invention sets like 1 operative condition. [when the important point verification parts which two or more parts are contained in said contents block data, and are the candidates for verification are plurality] As opposed to the connection data of the parts check value acquired by performing cipher processing with the application of the contents check value generation key for every parts It is characterized by making into a contents check value the result obtained by performing cipher processing which furthermore applied said contents check value generation key, and giving the generated contents check value to the contents data containing the contents block data for verification.

[0059] Furthermore, the 4th side face of this invention is a program offer medium

which offers the computer program which makes data processing which processes the contents data offered by a storage or communication media perform on computer system. Said computer program the step which generates a contents check value per contents block data for [which is contained in data] verification, and by performing collating processing of the generated contents check value It is in the program offer medium characterized by including the step which performs justification verification processing of the contents block data unit in said data.

[0060] The program offer medium concerning this invention is a medium which offers a computer program in a computer-readable format to the general purpose computer system which can perform various program codes, for example. Especially the gestalten, such as transmission media, such as storages, such as CD, and FD, MO, or a network, are not limited for a medium.

[0061] Such a program offer medium defines the collaboration-relation on the structure of the computer program and offer medium for realizing the function of a computer program predetermined in a computer system top, or a function. If it puts in another way, by installing a computer program in computer system through this offer medium, on computer system, a collaboration-operation is demonstrated and the same operation effectiveness as other side faces of this invention can be acquired.

[0062] The purpose, the description, and advantage of further others of this invention will become [rather than] clear by detailed explanation based on the example and the drawing to attach of this invention mentioned later.

[0063]

[Embodiment of the Invention] The gestalt of operation of this invention is explained below. The procedure of explanation is performed according to the following items.

In a data-processor configuration (2) contents data format (3) data processor (1) The storing data configuration (6) record regenerator of the storing data configuration (5) storage device of an applicable cipher-processing outline (4) record regenerator, The mutual recognition processing between storage devices The outline of mutual recognition processing (6-1) The contents data format of key message-exchange (10) plurality after the regeneration (9) mutual recognition in the record regenerator of the download processing (8) storage-device storing information from the change (7) record regenerator of the key block at the time of mutual recognition to a storage device, (6-2) The download corresponding to each format And regeneration (11) The check value in a content provider The program starting processing (15) contents configuration based on the starting priority in the handling plan in the control (14) contents data of the code reinforcement in cipher-processing key generation

configuration (13) cipher processing based on a generation processing mode (12) master key (ICV) And the exclusion (RIBOKESHON) configuration (18) secure chip configuration and the manufacture approach [0064] of generation of playback (expanding) processing (16) save data and storing in a storage device, and a regeneration (17) inaccurate device (1) The whole configuration block Fig. concerning one example of the data processor of this invention is shown in the data-processor block diagram 2 . The data processor of this invention uses the record regenerator 300 and a storage device 400 as main components.

[0065] The record regenerator 300 is constituted by a personal computer (PC:Personal Computer) or the game device. As the record regenerator 300 is shown in drawing 2 The control section 301 which performs generalization-control including communications control with the storage device 400 at the time of cipher processing in the record regenerator 300, the record regenerator cipher-processing section 302 which manages cipher processing at large, the storage device 400 connected to a record regenerator, and authentication processing It has the reading section 304 which reads data from the media 500 which perform and write data, such as the record device controller 303 and DVD, at least, and the exterior and the communications department 305 which performs transmission and reception of data.

[0066] The record regenerator 300 performs download of contents data to a storage device 400, and contents data playback from a storage device 400 by control of a control section 301. To the record regenerator 300, a storage device 400 is a desirable removable storage, for example, a memory card etc., and has external memory 402 constituted by RAM with nonvolatile memory, such as EEPROM and a flash memory, a hard disk, and a cell etc.

[0067] The record regenerator 300 has the communications department 305 as an interface which can input the contents data distributed from networks, such as the reading section 304 as an interface which can input the storage shown in the left end of drawing 2 , and the contents data stored in DVD, CD, FD, and HDD, and the Internet, and inputs contents from the exterior.

[0068] The record regenerator 300 has the cipher-processing section 302, and performs verification processing of data etc. to the authentication processing at the time of reproducing contents data from a storage device 400, and performing the contents data inputted from the outside through a read station 304 or the communications department 305 to a storage device 400, in case download processing is carried out, encryption processing, decryption processing, and a pan. The cipher-processing section 302 holds the information on the key the control

section 306 which controls the cipher-processing section 302 whole, and for cipher processing etc., and consists of a code / the decryption section 308 which performs the internal memory 307 and encryption processing in which processing was performed so that data could not be easily read from the exterior, decryption processing, generation and verification of the data for authentication, generating of a random number, etc.

[0069] An initialization instruction is transmitted to a storage device 400, or a control section 301 carries out the agency processing in various processings, such as mutual recognition processing performed through the record device controller 303 between the code / decryption section 308 of the record regenerator cipher-processing section 302, and the code / decryption section 406 of the storage device cipher-processing section 401, check value collating processing, encryption, and decryption processing, when the record regenerator 300 is equipped with a storage device 400. The latter part explains these the processings of each to a detail.

[0070] The cipher-processing sections 302 are authentication processing, encryption processing, decryption processing, and the processing section that performs verification processing of data etc. further as mentioned above, and have the cipher-processing control section 306, an internal memory 307, and the code / decryption section 308.

[0071] The authentication processing by which the cipher-processing control section 306 is performed in the record regenerator 300, It is the control section which performs control about cipher processing at large [, such as encryption/decryption processing,]. For example, a setup of the completion flag of authentication at the time of completion of the authentication processing performed between the record regenerator 300 and a storage device 400, Control about cipher processing at large, such as a run command of the various processings performed in the code / decryption section 308 of the record regenerator cipher-processing section 302, for example, download, and the check value generation processing about playback contents data and a run command of generation processing of various key data, is performed.

[0072] Although the latter part explains an internal memory 307 to a detail, it stores the key data which are needed in various processings, such as mutual recognition processing performed in the record regenerator 300, check value collating processing, encryption, and decryption processing, or discernment data.

[0073] A code / decryption section 308 use the key data stored in the internal memory 307, and in case download processing of the contents data inputted from the outside carries out at a storage device 400, processing of the authentication

processing at the time of reproducing and performing the contents data stored in the storage device 400 from a storage device 400, encryption processing, decryption processing, generation and verification of a further predetermined check value or electronic signature, verification of data, generating of a random number, etc. performs.

[0074] Here, since the internal memory 307 of the record regenerator cipher-processing section 302 holds important information, such as a cryptographic key, it is necessary to make it into the structure which is hard to read unjustly from the exterior. Therefore, the cipher-processing section 302 consists of semiconductor chips with the structure which is hard to access from the outside, it has multilayer structure and the width of face of the electrical potential difference or/which the memory of the interior is pinched by dummy layers, such as an aluminum layer, or is constituted by the lowest layer and operates, and a frequency is constituted unjustly [that it is narrow etc.] from the outside as Tampa-proof memory in which read-out of data has a difficult property. The latter part explains this configuration to a detail.

[0075] The record regenerator 300 is equipped with arithmetic and program control (Maine CPU:Central Processing Unit) 106, RAM (Random Access Memory)107 and ROM (Read Only Memory)108, AV processing section 109, the input interface 110, and PIO (parallel I/O interface)111 and SIO (serial I/O interface)112 other than these code processing facilities.

[0076] Arithmetic and program control (Maine CPU:Central Processing Unit) 106, and RAM (Random Access Memory)107 and ROM (Read Only Memory)108 are the configuration sections which function as a control system of record regenerator 300 body, and they function as the regeneration section which performs playback of the data decoded mainly in the record regenerator cipher-processing section 302. For example, arithmetic and program control (Maine CPU:Central Processing Unit) 106 performs control about playback of contents, such as outputting the contents data which were read from the storage device to the basis of control of a control section 301, and were decoded to AV processing section 109, and activation.

[0077] RAM107 is used as primary-storage memory for [various] processing in CPU106, and is used as a working area for processing by Maine CPU 106. A basic program for ROM108 to start OS started in Maine CPU 106 etc. is stored.

[0078] AV processing section 109 has data compression expanding processors, such as an MPEG 2 decoder, an ATRAC decoder, and an MP3 decoder, and, specifically, performs processing for the data output to data output devices, such as a display which was been [a display / it] attached or connected to the body of a record

regenerator and which is not illustrated, or a loudspeaker.

[0079] The input interface 110 outputs the input data from various kinds of input means, such as a connected controller, a keyboard, and a mouse, to Maine CPU 106. Maine CPU 106 performs processing which followed directions from the controller from a user based on the game program for example, under activation etc.

[0080] PIO (parallel I/O interface)111 and SIO (serial I/O interface)112 are used as a connection interface with storage, such as a memory card and a game cartridge, a portable electronic device, etc.

[0081] Moreover, Maine CPU 106 also performs control at the time of memorizing to a storage device 400 by using the setting data about the game for example, under activation etc. as save data. In the case of this processing, stored data is transmitted to a control section 301, and a control section 301 makes the cipher-processing section 302 perform cipher processing about save data if needed, and stores encryption data in a storage device 400. The latter part explains these cipher processing to a detail.

[0082] As mentioned above, preferably, to the record regenerator 300, a storage device 400 is a removable storage, for example, is constituted by the memory card. A storage device 400 has the cipher-processing section 401 and external memory 402.

[0083] The storage device cipher-processing sections 401 are the mutual recognition processing between the record regenerator 300 in the time of regeneration of download of the contents data from the record regenerator 300, or the contents data from the storage device 400 to the record regenerator 300 etc., and a storage device 400, encryption processing, decryption processing, and the processing section that performs verification processing of data etc. further, and have a control section, an internal memory, a code / decryption section, etc. like the cipher-processing section of the record regenerator 300. These details are shown in drawing 3 . As mentioned above, external memory 402 is constituted by the nonvolatile memory and the hard disk which consist of flash memories, such as EEPROM, RAM with a cell, etc., and stores the enciphered contents data.

[0084] Drawing 3 is drawing having shown the configuration with the record regenerator 300 which inputs contents focusing on the configuration about cipher processing in a storage device 400 from these contents offer means 500,600 while showing the outline of the media 500 which are contents offer means by which the data processor of this invention receives data supply, and the data configuration inputted from means of communications 600.

[0085] Media 500 are for example, optical disk media, magnetic-disk media, magnetic

tape media, semi-conductor media, etc. Means of communications 600 is a means in which data communication, such as the Internet communication link, a cable communication link, and satellite communication, is possible.

[0086] In drawing 3 , the record regenerator 300 verifies the media 500 which are contents offer means and the data inputted from means of communications 600, i.e., the contents according to a predetermined format as shown in drawing 3 , and saves contents after verification at a storage device 400.

[0087] As shown in the media 500 of drawing 3 , and means-of-communications 600 part, contents data have the following configuration sections.

Identification information: Identification information as an identifier of contents data.

Handling plan: A handling plan including the use limit information on the ability of other devices to use whether the configuration information of contents data, for example, the header size which constitutes contents data, contents section size, the version of a format, and contents can use only by the program, the contents type in which data etc. are shown, and the device which contents downloaded further etc.

Block information: Block information which consists of number of contents blocks, a block size, an encryption flag that shows the existence of encryption.

Key data: Key data which consist of an encryption key which enciphers above-mentioned block information, or a contents key which enciphers a contents block.

Contents block: The contents block which consists of the program data used as the actual candidate for playback, music, image data, etc.

It ****. In addition, the latter part explains a contents data detail to a detail further below using drawing 4 .

[0088] It is enciphered with a contents key (here, this is called a contents key (Content Key (hereafter referred to as Kcon))), and the record regenerator 300 is provided with contents data from media 500 and means of communications 600. Contents are storable in the external memory of a storage device 400 through the record regenerator 300.

[0089] For example, a storage device 400 enciphers the contents contained in contents data and the block information included as header information of contents data, and the various key information Kcon, for example, a contents key etc., using the key (here, this is called a preservation key (Storage Key (hereafter referred to as Kstr))) of the storage device proper stored in the internal memory 405 in a storage device, and memorizes it to external memory 402. In the download processing to a storage device 400 from the record regenerator 300 of contents data, or regeneration

of the contents data stored in the storage device 400 by the record regenerator 300, predetermined procedure, such as mutual recognition processing between devices, a contents data encryption, and decryption processing, is needed. The latter part explains these processings to a detail.

[0090] A storage device 400 has the cipher-processing section 401 and external memory 402, as shown in drawing 3 , and the cipher-processing section 401 has a control section 403, the communications department 404, an internal memory 405, the code / decryption section 406, and the external memory control section 407.

[0091] A storage device 400 interprets the command from the record regenerator 300, and serves as the storage device cipher-processing section 401 which performs processing from the external memory 402 holding contents etc. while it manages cipher processing at large and controls external memory 402.

[0092] The storage device cipher-processing section 401 holds the information on the control section 403 which controls the storage device cipher-processing section 401 whole, the communications department 404 which performs transmission and reception of the record regenerator 300 and data, the key data for cipher processing, etc. It has the code / decryption section 406 which performs the internal memory 405 and encryption processing in which processing was performed so that it could not read easily from the exterior, decryption processing, generation and verification of the data for authentication, generating of a random number, etc., and the external memory control section 407 which write the data of external memory 402.

[0093] The authentication processing by which a control section 403 is performed in a storage device 400, It is the control section which performs control concerning cipher processing at large [, such as encryption/decryption processing,]. For example, a setup of the completion flag of authentication at the time of completion of the authentication processing performed between the record regenerator 300 and a storage device 400, Control about cipher processing at large, such as a run command of the various processings performed in the code / decryption section 406 of the cipher-processing section 401, for example, download, and the check value generation processing about playback contents data and a run command of generation processing of various key data, is performed.

[0094] Although the latter part explains an internal memory 405 to a detail, it is constituted by memory with two or more blocks, and has the composition of having stored two or more groups, such as key data which are needed in various processings, such as mutual recognition processing performed in a storage device 400, check value collating processing, encryption, and decryption processing, or discernment data.

[0095] Like the internal memory 307 of the record regenerator cipher-processing section 302 explained previously, since the internal memory 405 of the storage device cipher-processing section 401 holds important information, such as a cryptographic key, it is necessary to make it into the structure which is hard to read unjustly from the exterior. Therefore, the cipher-processing section 401 of a storage device 400 consists of semiconductor chips with the structure which is hard to access from the outside, and it has multilayer structure, and considers as the configuration made into the property that read-out of data is difficult unjustly [that the width of face of the electrical potential difference or/which the memory of the interior is pinched by dummy layers, such as an aluminum layer or is constituted by the lowest layer and operates, and a frequency is narrow etc.] from the outside. In addition, the record regenerator cipher-processing section 302 may be the software constituted so that secret information, such as a key, might not be leaked outside easily.

[0096] A code / decryption section 406 Download processing of the contents data from the record regenerator 300, Regeneration of the contents data stored in the external memory 402 of a storage device 400, Or in the case of the mutual recognition processing between the record regenerator 300 and a storage device 400, the key data stored in the internal memory 405 are used, and processing of verification processing of data, encryption processing, decryption processing, generation and verification of a predetermined check value or electronic signature, generating of a random number, etc. is performed.

[0097] It connects with the record device controller 303 of the record regenerator 300, and the communications department 404 communicates the transfer data between the record regenerator 300 in the case of download processing of contents data, regeneration, or mutual recognition processing, and a storage device 400 according to control of the control section 301 of the record regenerator 300, or the control section 403 of a storage device 403.

[0098] (2) Explain the data format of the data which are stored in the media 500 in the system of this invention, or circulate the data communication means 600 top using a contents data format next drawing 4 thru/or drawing 6 .

[0099] The configuration shown in drawing 4 is drawing showing a format of the whole contents data, the configuration shown in drawing 5 is drawing showing the detail of the "handling plan" which constitutes a part of header unit of contents data, and the configuration shown in drawing 6 is drawing showing the detail of the "block information" which constitutes a part of header unit of contents data.

[0100] in addition, two or more different data formats, such as a format which was

suitable for real-time operations, such as a format corresponding to a game program, and music data, in the system of this invention, for example although here explained a typical example of the data format applied in the system of this invention, -- available -- the voice of these formats -- the latter part "(10) download corresponding to two or more contents data formats and each format and regeneration" is described in more detail like therefore.

[0101] In the data format shown in drawing 4 , the part shown in gray is enciphered data, and the parts of alteration check data and others with the white part of a double plate are data of the plaintext which is not enciphered. The encryption key of the encryption section is a key shown in the left of each frame. In the example shown in drawing 4 , what is not enciphered as what was enciphered by each block (contents block data) of the contents section is intermingled. These gestalten may be the configurations that all the contents block data that differ according to contents data and are contained in data are enciphered.

[0102] The data format is divided into a header and the contents section as shown in drawing 4 . A header Identification information (Content ID), a handling plan (Usage Policy), Check value A () [Integrity Check Value A] (It is hereafter referred to as ICVa), a block information key () [Block Information Table Key] (It is hereafter referred to as Kbit), the contents key Kcon, block information () [Block Information Table] It is constituted by (it is hereafter referred to as BIT), the check value B (ICVb), and the total check value (ICVt), and the contents section consists of two or more contents blocks (for example, enciphered contents and contents which are not enciphered).

[0103] Here, identification information shows the identifier according to individual for identifying contents (Content ID). The header size which shows the size for a header as a handling plan shows the detail to drawing 5 (Header Length), The contents size which shows the size of a contents part (Content Length), The format version which shows the version information of a format (Format Version), The format type in which the class of format is shown (Format Type), The contents type in which the class of contents [be / about whether the contents saved in the contents section are programs / it / data] is shown (Content Type), The starting priority information which shows starting priority in case a contents type is a program (Operation Priority), Whether the contents downloaded according to this format can use only by the downloaded device The use limit information which shows whether other same devices can be used (LocalizationField), The duplicate limit information which shows whether it is that the contents downloaded according to this format can reproduce

from the downloaded device to other same devices (Copy Permission), The movement restriction information which shows whether it is that the contents downloaded according to this format can move to other same devices from the downloaded device (Move Permission), The cryptographic algorithm which shows the algorithm used for carrying out the code of the contents block of contents circles (Encryption Algorithm), It consists of encryption mode (Encryption Mode) which shows the operation of the algorithm used for enciphering the contents of contents circles, and a verification approach (Integrity Check Method) which shows the generation method of a check value.

[0104] In addition, the data item recorded on the handling plan mentioned above is one example, and can record various handling plan information according to the mode of corresponding contents data. For example, although latter "exclusion (RIBOKESHON) configuration of (17) inaccurate devices" describes in detail, it is also possible to record the identifier of an inaccurate record regenerator as data, and to constitute so that the contents use by the inaccurate device may be eliminated by collating at the time of use initiation.

[0105] The check values A and ICVa are check values for verifying the alteration of identification information and a handling plan. It functions as the check value of the partial data instead of the whole contents data, i.e., a partial check value. It is used for the data block information key Kbit enciphering block information, and the contents key Kcon is used for enciphering a contents block. In addition, the block information key Kbit and the contents key Kcon are enciphered on media 500 and means of communications 600 with the delivery key (Distribution Key (hereafter referred to as Kdis)) mentioned later.

[0106] The detail of block information is shown in drawing 6 . In addition, the block information on drawing 6 is data altogether enciphered with the block information key Kbit as being understood from drawing 4 . Block information consists of contents block information on the contents block count (Block Number) and N individual which shows the number of contents blocks, as shown in drawing 6 . Contents block information consists of the block size (Block Length), an encryption flag (Encryption Flag) which shows whether it is enciphered or not, a flag for verification (ICV Flag) which shows whether it is necessary to calculate a check value, and a contents check value (ICVi).

[0107] A contents check value is a check value used in order to verify the alteration of each contents block. The column of two or more latter data formats, and "the download processing to the storage device corresponding to each format and

regeneration from a storage device" explains the example of the generation technique of a contents check value. [(10)] In addition, the block information key Kbit which has enciphered block information is further enciphered with the delivery key Kdis.

[0108] Explanation of the data format of drawing 4 is continued. The check values B and ICVb are check values for verifying the alteration of the block information key Kbit, the contents key Kcon, and block information. It functions as the check value of the partial data instead of the whole contents data, i.e., a partial check value. The total check value ICVt is a check value for verifying the alteration of all the data used as the check values ICVi of ICVa, ICVb, and each contents block (when set up), these partial check values, or the candidate for a check of those.

[0109] In addition, in drawing 6 , although it enables it to set up freely a block size, an encryption flag, and the flag for verification, it is good also as a configuration which determined the Ruhr to some extent. For example, a cipher field and a plaintext field may be made into a fixed size repeat, or all contents data may be enciphered, and the block information BIT may be compressed. Moreover, in order to make the contents key Kcon differ for every contents block, you may make it include the contents key Kcon in the contents block instead of a part for a header. The example of a contents data format is further explained to a detail in the item of "the download and regeneration" corresponding to two or more contents data formats and each format. [(10)]

[0110] (3) the voice of various cipher processing which may be applied in an applicable cipher-processing outline, next the data processor of this invention in the data processor of this invention -- attach like and explain. in addition, the various processings in the data processor of this invention which explains concretely the explanation about cipher processing shown in outline" of applicable cipher processing in the data processor of this paragraph eye "(3) this invention in the latter part, for example, the authentication processing between a. record regenerator and a storage device. b. Download processing to the storage device of contents. c. the voice of cipher processing used as the foundation of the processing performed in processing of regeneration of the contents stored in the storage device etc. -- attach like and explain the outline. The concrete processing in the record regenerator 300 and a storage device 400 is explained below to the item (4) of this specification for every processing at a detail.

[0111] In the following and a data processor about the outline of applicable cipher processing The message authentication by the common key encryption system (3-1) (3-2) The mutual recognition (3-7) elliptic curve cryptosystem by the mutual

recognition (3-5) public key certificate (3-6) public key cryptosystem by the verification (3-4) common key encryption system of the electronic signature by the electronic signature (3-3) public key cryptosystem by the public key cryptosystem It explains in order of the decryption processing (3-9) random-number generation processing using the used encryption processing (3-8) elliptic curve cryptosystem.

[0112] (3-1) message **** by the common key encryption system -- generation processing of the alteration detection data using a common key encryption system is explained first. Alteration detection data are data for attaching to the data which want to detect an alteration and carrying out the check and implementer authentication of an alteration.

[0113] For example, each check values A and B for a duplex frame part in the DS explained by drawing 4 , the total check value, the contents check value stored in each block in the block information shown in drawing 6 are generated as this alteration detection data.

[0114] Here, the example using DES in a common key encryption system is explained as one of the examples of the generation art of electronic signature data. In addition, in this invention, it is also possible to use FEAL (Fast Encipherment ALgorithm:NTT), AES (Advanced Encryption Standard: the U.S. next standard code), etc. as processing in the same common key encryption system besides DES.

[0115] The generation method of electronic signature using general DES is explained using drawing 7 . First, it precedes generating electronic signature and the message set as the object of electronic signature is divided per 8 bytes (the divided message is hereafter set to M1, M2, ..., MN). And the exclusive OR of M1 is carried out to initial value (Initial Value (hereafter referred to as IV)) (the result is set to I1). Next, I1 is put into the DES encryption section, and it enciphers using a key (hereafter referred to as K1) (an output is set to E1). Continuously, the exclusive OR of E1 and M2 is carried out, the output I2 is put in to the DES encryption section, and it enciphers using a key K1 (output E2). Hereafter, this is repeated and encryption processing is performed to all messages. EN which came out at the end becomes electronic signature. Generally, this value is called a message authenticator (MAC (Message Authentication Code)), and is used for the alteration check of a message. Moreover, the thing of the method to which the chain of the cipher is carried out in this way is called the CBC (Cipher Block Chaining) mode.

[0116] In addition, the MAC value outputted in an example of generation like drawing 7 is usable as each check values A and B for a duplex frame part in the DS shown by drawing 4 , the total check value and the contents check value ICV1 stored in each

block in the block information shown in drawing 6 – ICVN. At the time of verification of this MAC value, when a verification person generates a MAC value by the same approach as a generate time and the same value is acquired, it considers as a verification success.

[0117] In addition, although the exclusive OR of the initial value IV was carried out to the first 8-byte message M1 in the example shown in drawing 7 , it is also possible to consider as the configuration which does not carry out the exclusive OR of the initial value as initial value IV=0.

[0118] The processing block diagram showing the MAC value generation method which raised security further is shown in drawing 8 to the MAC value generation method shown in drawing 7 . Drawing 8 shows the example which replaces with the single DES of drawing 7 and performs generation of a MAC value using Triple DES (Triple DES).

[0119] The example of a detail configuration of each Triple DES (Triple DES) configuration section shown in drawing 8 is shown in drawing 9 . As shown in drawing 9 (a) and (b), there are two different modes in the configuration as Triple DES (Triple DES). Drawing 9 (a) shows the example which used two cryptographic keys, and processes in order of the encryption processing with a key 1, the decryption processing with a key 2, and encryption processing according to a key 1 further. A key is used two kinds in order of K1, K2, and K1. Drawing 9 (b) shows the example which used three cryptographic keys, processes in order of the encryption processing with a key 1, the encryption processing with a key 2, and encryption processing according to a key 3 further, and performs encryption processing 3 times. Three kinds of keys are used for a key in order of K1, K2, and K3. Thus, by considering as the configuration which two or more processings are made to follow, security reinforcement is raised as compared with Single DES. However, this Triple DES (Triple DES) configuration has the fault whose processing time is Single DES that it takes about 3 times.

[0120] The example of a MAC value generation configuration which improved the Triple DES configuration explained by drawing 8 and drawing 9 is shown in drawing 10 . In drawing 10 , all encryption processings to each message from the start of the message train used as the candidate for a signature to the middle are considered as processing by Single DES, and are considered as the Triple DES (Triple DES) configuration which shows only the encryption processing to the last message to drawing 9 (a).

[0121] By considering as such a configuration shown in drawing 10 , the generation processing time of the MAC value of a message becomes possible [it being shortened

almost to the same extent as the time amount which the MAC value generation processing by Single DES takes, and raising security rather than the MAC value by Single DES]. In addition, the Triple DES configuration to the last message can also be considered as the configuration of drawing 9 (b).

[0122] (3-2) Although beyond the electronic signature by the public key cryptosystem is the generation method of the electronic signature data at the time of applying a common key encryptosystem-ized method as a cipher system next, explain the generation method of electronic signature using the public key cryptosystem as a cipher system using drawing 11 . The processing shown in drawing 11 is the generation processing flow of the electronic signature data which used EC-DSA (Elliptic Curve Digital Signature Algorithm) (IEEE P1363/D3). In addition, the example which used the elliptic curve cryptosystem (Elliptic Curve Cryptography (hereafter referred to as ECC)) as public key encryption here is explained. in addition, in the data processor of this invention, it is also possible to use RSA cryptograph (Rivest, Shamir, Adleman), such as etc. (ANSI X9.31), in the same public key cryptosystem besides an elliptic curve cryptosystem.

[0123] Each step of drawing 11 is explained. In step S1, let the base point on an elliptic curve, and r into the order of G, and let [p / the characteristic, and a and b] Ks be a private key (0<Ks<r) for the multiplier (elliptic curve: y2=x3+ax+b) of an elliptic curve, and G. Step S2 The hash value of Message M is calculated by setting, and it considers as f=Hash (M).

[0124] Here, how to calculate a hash value using a Hash Function is explained. A Hash Function is a function which considers a message as an input, compresses this into the data of predetermined bit length, and is outputted as a hash value. It is difficult for a Hash Function to predict an input from a hash value (output), and when 1 bit of the data inputted into the Hash Function changes, discovering different input data which many bits of a hash value change and has the same hash value has the difficult description. As a Hash Function, MD4, MD5, SHA-1, etc. may be used and same DES-CBC may be used with drawing 7 etc. having explained. In this case, MAC (check value: it is equivalent to ICV) used as a final output value serves as a hash value.

[0125] Continuously, at step S3, a random number u (0<u<r) is generated and the coordinate V (Xv, Yv) which doubled the base point u by step S4 is calculated. In addition, the addition on an elliptic curve and 2 double ** are defined as follows.

[0126]

[Equation 1]

P=(Xa, Ya),Q=(Xb, Yb),R=(Xc,Yc)=P+Qとすると、
P≠Qの時 (加算)、
$Xc=\lambda^2-Xa-Xb$
$Yc=\lambda\times(Xa-Xc)-Ya$
$\lambda=(Yb-Ya)/(Xb-Xa)$
P=Qの時 (2倍算)、
$Xc=\lambda^2-2Xa$
$Yc=\lambda\times(Xa-Xc)-Ya$
$\lambda=(3(Xa)^2+a)/(2Ya)$

[0127] u times of Point G are calculated using these (although a rate is slow, it carries out as follows as the most intelligible operation approach.). G, 2xG, and 4xG .. is calculated and 2 ixG (value which 2-double-**(ed) G i times) corresponding to the place carries out binary number expansion of the u, and 1 stands is added (bit position when counting i from LSB of u).

[0128] At step S5, c=Xvmod r is calculated and it judges whether this value is set to 0 at step S6, if it is not 0, d=[(f+cKs) /u] mod r will be calculated at step S7, it judges whether d is 0 at step S8, and if d is not 0, c and d will be outputted as electronic signature data by step S9. If r is assumed to be the die length of 160 bit length, electronic signature data serve as 320 bit length.

[0129] In step S6, when c is 0, it returns to step S3 and a new random number is regenerated. Similarly, when d is 0 at step S8, it returns to step S3 and a random number is regenerated.

[0130] (3-3) Explain verification of the electronic signature by the public key cryptosystem, next the verification approach of electronic signature using a public key cryptosystem using drawing 12 . step S11 -- M -- let the multiplier (elliptic curve: y2=x3+ax+b) of an elliptic curve, and G as the base point on an elliptic curve, and let [a message and p / the characteristic, and a and b] the order of G, G, and KsxG be public keys (0<Ks<r) for r. It verifies whether the electronic signature data c and d fill 0<c<r and 0<d<r with step S12. When this is being filled, at step S13, the hash value of Message M is calculated and it considers as f=Hash (M). Next, h= 1-/d mod r is calculated at step S14, and it is h1=fh at step S15. mod r, h2=ch mod r is calculated.

[0131] In step S16, point P=(Xp, Yp) =h1xG+h2 and KsxG are calculated using h1 and h2 which were already calculated. Since the electronic signature verification person knows a public key G and KsxG, he can do count of the scalar multiple of the point on an elliptic curve like step S4 of drawing 11 . And Point P judges whether it is an infinite point at step S17, and if it is not an infinite point, it will progress to step S18 (the judgment of an infinite point will be able to be performed at step S16 in fact.). That is,

if addition of P= (X, Y) and Q= (X, −Y) is performed, lambda cannot be calculated but it will have become clear that P+Q is an infinite point. Xp mod r is calculated at step S18, and it compares with the electronic signature data c. Finally, when this value is in agreement, it progresses to step S19 and electronic signature judges with the right.

[0132] When electronic signature is judged to be the right, it turns out that data were not altered but the person holding the private key corresponding to a public key generated electronic signature.

[0133] In step S12, when the electronic signature data c or d do not fill 0<c<r and 0<d<r, it progresses to step S20. Moreover, in step S17, also when Point P is an infinite point, it progresses to step S20. In step S18, also when the value of Xp mod r is not in agreement with the electronic signature data c, it progresses to step S20 further again.

[0134] In step S20, when judged with electronic signature not being right, it turns out that those who data are altered or hold the private key corresponding to a public key did not generate electronic signature.

[0135] (3-4) The mutual recognition approach using the mutual recognition, next the common key encryption system by the common key encryption system is explained using drawing 13 . In drawing 13 , although DES is used as a common key encryption system, as long as it is the common key encryption system same as mentioned above, any are sufficient. In drawing 13 , the random number Rb whose B is 64 bits is generated first, and ID (b) which is Rb and self ID is transmitted to A. A which received this newly generates the 64-bit random number Ra, in order of Ra, Rb, and ID (b), Key Kab is used for it in the CBC mode of DES, it enciphers data, and returns them to B. According to the CBC mode processing configuration of DES shown in drawing 7 , M1 and Rb are equivalent to M2, Ra is [ID (b)] equivalent to M3, and the outputs E1, E2, and E3 when being referred to as initial value:IV=0 serve as a cipher.

[0136] B which received this decrypts received data with Key Kab. First, the decryption approach of received data decrypts a cipher E1 with Key Kab, and obtains a random number Ra. Next, a cipher E2 is decrypted with Key Kab, the exclusive OR of E1 is carried out to the result, and Rb is obtained. Finally, a cipher E3 is decrypted with Key Kab, the exclusive OR of E2 is carried out to the result, and ID (b) is obtained. In this way, Rb and ID (b) verify whether it is in agreement with what B transmitted among Ra, Rb(s), and ID (b) which were obtained. When it passes in this verification, B attests A as a just thing.

[0137] Next, B generates the session key (Session Key (hereafter referred to as Kses)) used after authentication (a random number is used for a generation method).

And in order of Rb, Ra, and Kses, in the CBC mode of DES, Key Kab is used, it enciphers, and A is returned.

[0138] A which received this decrypts received data with Key Kab. Since the decryption approach of received data is the same as that of decryption processing of B, a detail is omitted here. In this way, Rb and Ra verify whether it is in agreement with what A transmitted among Rb(s), Ra, and Kses(es) which were obtained. When it passes in this verification, A attests B as a just thing. After attesting the partner of each other, the session key Kses is used as a common key for the secret communication after authentication.

[0139] In addition, when injustice and an inequality are found on the occasion of verification of received data, processing is interrupted as that in which mutual recognition failed.

[0140] (3-5) Explain a public key certificate, next a public key certificate using drawing 14 . A public key certificate is a certificate which the certificate authority (CA:Certificate Authority) in a public key cryptosystem publishes, and when a user submits self ID, a public key, etc. to a certificate authority, it is a certificate with which a certificate authority side adds information, such as ID of a certificate authority, and an expiration date, adds the signature by the certificate authority further, and is created.

[0141] The public key certificate shown in drawing 14 includes electronic signature in the algorithm used for the version number of a certificate, the serial number of the certificate which a certificate authority assigns to a certificate user, and electronic signature and a parameter, the identifier of a certificate authority, the expiration date of a certificate, a certificate user's identifier (user ID), and a certificate user's public key list.

[0142] Electronic signature is data which generated the hash value with the application of the Hash Function to a certificate user's whole public key in the algorithm used for the version number of a certificate, the serial number of the certificate which a certificate authority assigns to a certificate user, and electronic signature and the parameter, the identifier of a certificate authority, the expiration date of a certificate, and a certificate user's identifier list, and were generated using the private key of a certificate authority to the hash value. The processing flow explained by drawing 11 is applied to generation of this electronic signature.

[0143] a certificate authority updates the public key certificate with which the expiration date went out, and performs creation of the inaccurate person list of [for excluding the user who performed injustice], management, and distribution (this --

RIBOKESHON: -- referred to as Revocation) while it publishes the public key certificate shown in drawing 14 . Moreover, generation of a public key and a private key is also performed if needed.

[0144] On the other hand, in case this public key certificate is used, using the public key of the certificate authority which self holds, a user verifies the electronic signature of the public key certificate concerned, after he succeeds in verification of electronic signature, he picks out a public key from a public key certificate, and uses the public key concerned. Therefore, all the users using a public key certificate need to hold the public key of a common certificate authority. In addition, about the verification approach of electronic signature, since drawing 12 explained, the detail is omitted.

[0145] (3-6) Explain the mutual recognition approach using the elliptic curve cryptosystem of the 160 bit length which is the mutual recognition, next the public key cryptosystem by the public key cryptosystem using drawing 15 . In drawing 15 , although ECC is used as a public key cryptosystem, as long as it is the public key cryptosystem same as mentioned above, any are sufficient. Moreover, key size may not be 160 bits, either. In drawing 15 , first, B generates the 64-bit random number Rb, and transmits to A. A which received this newly generates the 64-bit random number Ra and the random number Ak smaller than Characteristic p. And point Av=AkxG which doubled the base point G Ak is calculated, electronic signature A.Sig to Ra, Rb, and Av (X coordinate and Y coordinate) is generated, and B is returned with the public key certificate of A. Here, since 64 bits, and the X coordinate and Y coordinate of Av are 160 bits, Ra and Rb generate the electronic signature to a total of 448 bits, respectively. Since drawing 11 explained the generation method of electronic signature, the detail is omitted. Moreover, since drawing 14 also explained the public key certificate, the detail is omitted.

[0146] It verifies whether B of Rb which A has transmitted which received the public key certificate of A, Ra, Rb and Av, and electronic signature A.Sig corresponds with what B generated. Consequently, when in agreement, the electronic signature in the public key certificate of A is verified with the public key of a certificate authority, and the public key of A is taken out. Since verification of a public key certificate was explained using drawing 14 , the detail is omitted. And electronic signature A.Sig is verified using the taken-out public key of A. Since drawing 12 explained the verification approach of electronic signature, the detail is omitted. After succeeding in verification of electronic signature, B attests A as a just thing.

[0147] Next, B generates the random number Bk smaller than Characteristic p. And

point Bv=BkxG which doubled the base point G Bk is calculated, electronic signature B.Sig to Rb, Ra, and Bv (X coordinate and Y coordinate) is generated, and A is returned with the public key certificate of B.

[0148] It verifies whether A of Ra which B has transmitted which received the public key certificate of B, Rb, Ra and Av, and electronic signature B.Sig corresponds with what A generated. Consequently, when in agreement, the electronic signature in the public key certificate of B is verified with the public key of a certificate authority, and the public key of B is taken out. And electronic signature B.Sig is verified using the taken-out public key of B. After succeeding in verification of electronic signature, A attests B as a just thing.

[0149] When both succeed in authentication, B calculates BkxAv (although Bk is a random number, since Av is a point on an elliptic curve, scalar multiple count of the point on an elliptic curve is the need), and A calculates AkxBv, and after using 64 bits of low order of the X coordinate of these points as a session key, it is used for a communication link (when a common key encryptosystem is made into the common key encryptosystem of 64-bit key length). Of course, a session key may be generated from Y coordinate and you may not be 64 bits of low order. In addition, transmit data is not only enciphered with a session key, but electronic signature may be attached in the secret communication after mutual recognition.

[0150] When injustice and an inequality are found on the occasion of verification of electronic signature, or verification of received data, processing is interrupted as that in which mutual recognition failed.

[0151] (3-7) Explain the encryption processing using an elliptic curve cryptosystem, next the encryption using an elliptic curve cryptosystem using drawing 16 . step S21 -- setting -- Mx and My -- let the multiplier (elliptic curve: $y2=x3+ax+b$) of an elliptic curve, and G as the base point on an elliptic curve, and let [a message and p / the characteristic, and a and b] the order of G, G, and KsxG be public keys ($0<Ks<r$) for r. At step S22, a random number u is generated so that it may become $0<u<r$, and the coordinate V which doubled public key KsxG u at step S23 is calculated. In addition, since step S4 of drawing 11 explained the scalar multiple on an elliptic curve, it omits for details. At step S24, the X coordinate of V is doubled Mx, in quest of a remainder, it is referred to as X0 by p, the Y coordinate of V is doubled My at step S25, and it is referred to as Y0 in quest of a remainder by p. In addition, when there is less die length of a message than the number of bits of p, My uses a random number and cancels My in the decryption section. In step S26, uxG is calculated and cipher uxG, and (X0, Y0) are obtained at step S27.

[0152] (3-8) Explain the decryption processing using an elliptic curve cryptosystem, next the decryption using an elliptic curve cryptosystem using drawing 17 . step S31 -- setting -- uxG, and (X0, Y0) -- let the base point on an elliptic curve, and r into the order of G, and let [cipher data and p / the characteristic, and a and b] Ks be a private key (0<Ks<r) for the multiplier (elliptic curve: y2=x3+ax+b) of an elliptic curve, and G. In step s32, code data uxG is doubled private key Ks, and Coordinate V (Xv, Yv) is searched for. At step S33, the X coordinate of (X0, Y0) is taken out among code data, X1=X0-/Xv mod p is calculated, Y coordinate is taken out in step S34, and Y1=Y0-/Yv mod p is calculated. And Xone is set to Mx at step S35, and a message is taken out by setting Y1 to My. When My is not made into the message at this time, Y1 cancels.

[0153] Thus, the key used for encryption and the key used for a decryption can be used as a different key by making a private key to Ks and making a public key into G and KsxG.

[0154] Moreover, detailed explanation is omitted although RSA cryptograph is known as other examples of public key encryption (the detail is described by PKCS #1 Version2).

[0155] (3-9) Explain random-number generation processing, next the generation method of a random number. As a generation method of a random number, thermal noise is amplified and the intrinsic random-number generating method generated from the A/D output, the pseudo-random number generating method generated combining linear circuits, such as an M sequence, two or more are learned. Moreover, how to generate using common key encryptosystems, such as DES, is also learned. This example explains the pseudo-random number generation method which used DES (ANSI X9.17 base).

[0156] First, the kind (Seed) for Kr and random number generation is set to S for the key information used for D and Triple-DES in the value of 64 bits (a high order bit is set to 0 in the case of the number of bits not more than this) acquired from data, such as time amount. At this time, a random number R is calculated as follows.

[0157]

[Equation 2]

$$I = \text{Triple-DES}(Kr, D) \quad \cdots\cdots (2-1)$$

$$R = \text{Triple-DES}(Kr, S \hat{} I) \quad \cdots\cdots (2-2)$$

$$S = \text{Triple-DES}(Kr, R \hat{} I) \quad \cdots\cdots (2-3)$$

[0158] Here, Triple-DES() shall make the 1st argument cryptographic key information, the value of the 2nd argument shall be made into the function enciphered by

Triple-DES, and the exclusive OR of 64 bitwises and the value S which finally came out shall be updated for operation ^ as new Seed (seed).

[0159] Hereafter, in generating a random number continuously, it shall repeat a formula (2-2) and a formula (2-3).

[0160] In the above, in the data processor of this invention, the various processing modes about applicable cipher processing were explained. Next, the concrete processing performed in the data processor of this invention is explained to a detail.

[0161] (4) The storing data block diagram 18 of a record regenerator is drawing explaining the contents of data-hold of the internal memory 307 constituted by the record regenerator cipher-processing section 302 in the record regenerator 300 shown by drawing 3 .

[0162] As shown in drawing 18 , the following keys and data are stored in the internal memory 307.

MKake: The master key for storage device authentication keys for generating an authentication key (Authentication and Key Exchange Key (hereafter referred to as Kake)) required for the mutual recognition processing performed between the record regenerator 300 and a storage device 400 (refer to drawing 3).

IVake: Initial value for storage device authentication keys.

MKdis: The master key for delivery keys for generating the delivery key Kdis.

IVdis: Initial value for delivery key generation.

Kicva: The check value A generation key which is a key for generating the check value ICVa.

Kicvb: The check value B generation key which is a key for generating the check value ICVb.

Kicvc: The contents check value generation key which is a key for generating the check value ICVi of each contents block (i= 1 - N).

Kicvt: The total check value generation key which is a key for generating the total check value ICVt.

Ksys: The system signature key used since a signature or ICV common to a distribution system is attached.

Kdev: The record regenerator signature key of the record regenerator proper used since it differs for every record regenerator and a record regenerator attaches a signature or ICV.

IVmem: Initial value used for cipher processing in the cases, such as initial value and mutual recognition processing. As common as a storage device.

These keys and data are stored in the internal memory 307 constituted by the record

regenerator cipher-processing section 302.

[0163] (5) The storing data block diagram 19 of a storage device is drawing showing the data-hold situation on a storage device. In drawing 19 , the internal memory 405 is divided into two or more blocks (this example N block), and the following keys and data are stored during each block.

IDmem: Storage device identification information, identification information of a storage device proper.

Kake: An authentication key, the authentication key used at the time of mutual recognition with the record regenerator 300.

IVmem: Initial value used for cipher processing in the cases, such as initial value and mutual recognition processing.

Kstr: The cryptographic key of contents data besides a preservation key and a block information key.

Kr:random-number generation key and S:kind -- the data of these are respectively held to the block according to individual. External memory 402 holds the contents data of plurality (this example M pieces), and holds the data explained by drawing 4 , respectively like drawing 26 or drawing 27 . The latter part explains the difference in the configuration of drawing 26 and drawing 27 .

[0164] (6) The schematic diagram 20 of the mutual recognition processing (6-1) mutual recognition processing between a record regenerator and a storage device is a flow chart showing the authentication procedure of the record regenerator 300 and a storage device 400. In step S41, a user inserts a storage device 400 in the record regenerator 300. However, when using the storage device which can communicate by non-contact, it is not necessary to insert.

[0165] If a storage device 400 is set to the record regenerator 300, the storage device detection means in the record regenerator 300 shown in drawing 3 (not shown) will notify wearing of a storage device 400 to a control section 301. Next, in step S42, the control section 301 of the record regenerator 300 transmits an initialization instruction to a storage device 400 through the record device controller 303. In the control section 403 of the storage device cipher-processing section 401, the storage device 400 which received this receives an instruction through the communications department 404, and if the completion flag of authentication is set, it will clear it. That is, it is set as the condition of not attesting.

[0166] Next, in step S43, the control section 301 of the record regenerator 300 transmits an initialization instruction to the record regenerator cipher-processing section 302. At this time, a storage device insertion opening number is also

transmitted collectively. By transmitting a storage device insertion opening number, even if it is the case where two or more storage devices are connected to the record regenerator 300, authentication processing with two or more storage devices 400 and data transmission and reception are attained at coincidence.

[0167] In the control section 306 of the record regenerator cipher-processing section 302, the record regenerator cipher-processing section 302 of the record regenerator 300 which received the initialization instruction will be cleared, if the completion flag of authentication corresponding to a storage device insertion opening number is set. That is, it is set as the condition of not attesting.

[0168] Next, in step S44, the control section 301 of the record regenerator 300 specifies the key block number which the storage device cipher-processing section 401 of a storage device 400 uses. In addition, about the detail of a key block number, it mentions later. In step S45, the control section 301 of the record regenerator 300 reads the storage device identification information IDmem stored in the key block with which the internal memory 405 of a storage device 400 was specified. The control section 301 of the record regenerator 300 transmits the storage device identification information IDmem to the record regenerator cipher-processing section 302, and makes the authentication key Kake based on the storage device identification information IDmem generate in step S46. As a generation method of the authentication key Kake, it generates as follows, for example.

[0169]

[Equation 3]

$$Kake=DES(MKake, IDmem \hat{} IVake)$$

[0170] MKake is a master key for storage device authentication keys for generating the authentication key Kake required for the mutual recognition processing performed between the record regenerator 300 and a storage device 400 (refer to drawing 3) here, and this is a key stored in the internal memory 307 of the record regenerator 300 as mentioned above. Moreover, IDmem is the storage device identification information of a proper at a storage device 400. Furthermore, IVake is the initial value for storage device authentication keys. Moreover, in the above-mentioned formula, DES() makes the 1st argument a cryptographic key, it is the function which enciphers the value of the 2nd argument by DES, and operation ^ shows the exclusive OR of 64 bitwises.

[0171] For example, in applying the DES configuration shown in drawing 7 and drawing 8 , drawing 7 and the message M shown in 8 are set to storage device identification

information:IDmem, a key K1 is set to master key:MKake for device authentication keys, and the output obtained considering initial value IV as :IVake serves as the authentication key Kake.

[0172] Next, generation processing of mutual recognition and the session key Kses is performed at step S47. Mutual recognition is performed between the code / decryption section 308 of the record regenerator cipher-processing section 302, and the code / decryption section 406 of the storage device cipher-processing section 401, and the control section 301 of the record regenerator 300 is performing the agency.

[0173] Mutual recognition processing can be performed according to the processing explained by above-mentioned drawing 13 . In the configuration shown in drawing 13 , A and B correspond to the record regenerator 300 and a storage device 400, respectively. First, the record regenerator cipher-processing section 302 of the record regenerator 300 generates a random number Rb, and transmits the record regenerator identification information IDdev which is a random number Rb and self ID to the storage device cipher-processing section 401 of a storage device 400. In addition, the record regenerator identification information IDdev is the identifier of the regenerator proper memorized by the storage section constituted in the record regenerator 300. It is good also as a configuration which records the record regenerator identification information IDdev into the internal memory of the record regenerator cipher-processing section 302.

[0174] The storage device cipher-processing section 401 of a storage device 400 which received a random number Rb and the record regenerator identification information IDdev newly generates the 64-bit random number Ra, in order of Ra, Rb, and the record regenerator identification information IDdev, the authentication key Kake is used for it in the CBC mode of DES, it enciphers data, and returns them to the record regenerator cipher-processing section 302 of the record regenerator 300. For example, according to the CBC mode processing configuration of DES shown in drawing 7 , M1 and Rb are equivalent to M2, Ra is [IDdev] equivalent to M3, and the outputs E1, E2, and E3 when considering as initial value:IV=IVmem serve as a cipher.

[0175] The record regenerator cipher-processing section 302 of the record regenerator 300 which received ciphers E1, E2, and E3 decrypts received data with the authentication key Kake. First, the decryption approach of received data decrypts a cipher E1 with the authentication key Kake, carries out the exclusive OR of the result and IVmem, and obtains a random number Ra. Next, a cipher E2 is decrypted with the authentication key Kake, the exclusive OR of E1 is carried out to the result,

and Rb is obtained. Finally, a cipher E3 is decrypted with the authentication key Kake, the exclusive OR of E2 is carried out to the result, and the record regenerator identification information IDdev is obtained. In this way, Rb and the record regenerator identification information IDdev verify whether it is in agreement with what the record regenerator 300 transmitted among Ra and Rb which were obtained, and the record regenerator identification information IDdev. When it passes in this verification, the record regenerator cipher-processing section 302 of the record regenerator 300 attests a storage device 400 as a just thing.

[0176] Next, the record regenerator cipher-processing section 302 of the record regenerator 300 generates the session key (Session Key (hereafter referred to as Kses)) used after authentication (a random number is used for a generation method). And in order of Rb, Ra, and Kses, in the CBC mode of DES, Key Kake and initial value IVmem are used, it enciphers, and the storage device cipher-processing section 401 of a storage device 400 is returned.

[0177] The storage device cipher-processing section 401 of a storage device 400 which received this decrypts received data with Key Kake. Since the decryption approach of received data is the same as that of the decryption processing in the record regenerator cipher-processing section 302 of the record regenerator 300, a detail is omitted here. In this way, Rb and Ra verify whether it is in agreement with what the storage device 400 transmitted among Rb(s), Ra, and Kses(es) which were obtained. When it passes in this verification, the storage device cipher-processing section 401 of a storage device 400 attests the record regenerator 300 as a just thing. After attesting the partner of each other, the session key Kses is used as a common key for the secret communication after authentication.

[0178] In addition, when injustice and an inequality are found on the occasion of verification of received data, processing is interrupted as that in which mutual recognition failed.

[0179] When it succeeds in mutual recognition, while progressing to step S49 from step S48 and holding the session key Kses in the record regenerator cipher-processing section 302 of the record regenerator 300, the completion flag of authentication which shows that mutual recognition was completed is set. Moreover, when mutual recognition goes wrong, it progresses to step S50, and while canceling the session key Kses generated by the authentication processing process, the completion flag of authentication is cleared. In addition, when already cleared, clear processing is not necessarily required.

[0180] in addition, when a storage device 400 is removed from storage device

insertion opening The storage device detection means in the record regenerator 300 notifies that the storage device 400 was removed by the control section 301 of the record regenerator 300. The control section 301 of the record regenerator 300 which received this It orders to clear the completion flag of authentication corresponding to a storage device insertion opening number to the record regenerator cipher-processing section 302 of the record regenerator 300. The record regenerator cipher-processing section 302 of the record regenerator 300 which received this clears the completion flag of authentication corresponding to a storage device insertion opening number.

[0181] In addition, although the example which performs mutual recognition processing here according to the procedure shown in drawing 13 was explained, processing according to the mutual recognition procedure of not only the example of authentication processing mentioned above but drawing 15 explained previously, for example may be performed. Moreover, in the procedure shown in drawing 13 , A of drawing 13 may be used as the record regenerator 300, B may be used as a storage device 400, and mutual recognition processing may be performed for ID which the B:storage device 400 sends to A:record regenerator 300 first as storage device identification information under key block in a storage device. The authentication processing procedure performed in this invention can apply various processings, and is not limited to above-mentioned authentication processing.

[0182] (6-2) One description in the mutual recognition processing in the data processor of change this invention of the key block at the time of mutual recognition is the point of constituting two or more key blocks (key block of an ex.N individual), and the record regenerator 300 specifying one key block as a storage device 400 side (step S44 in the processing flow of drawing 20), and performing authentication processing. As previously explained in drawing 19 , two or more key blocks are formed in the internal memory 405 constituted by the cipher-processing section 401 of a storage device 400, and various data, such as key data, ID information, etc. that each differs, are stored. Mutual recognition processing performed between the record regenerator 300 explained by drawing 20 and a storage device 400 is performed to one key block of a key block of the plurality of the storage device 400 of drawing 19 .

[0183] The key used for mutual recognition with the configuration which performs mutual-recognition processing between a storage and its playback device conventionally: As for the authentication key, it was common that a common thing was used. When it is going to follow, for example, is going to change an authentication key for every product destination (according to country), and every product, it is

necessary to change the key data which are needed for the authentication processing by the side of a record regenerator and a storage device in both devices. The key data which are needed for the authentication processing stored in the record regenerator which followed, for example, was newly put on the market are not equivalent to the key data which are needed for the authentication processing stored in the storage device sold previously, and the situation access to the storage device of an old version becomes impossible generates a new record regenerator. Conversely, the same situation occurs also in the relation between the storage device of a high version, and the record regenerator of an old version.

[0184] In the data processor of this invention, as shown in drawing 19 , the key block as a key set with which plurality differs is beforehand stored in the storage device 400. The key block which should apply a record regenerator to authentication processing for every every product destination (according to country), a product, model, version, and application, i.e., an assignment key block, is set up. This setting information is stored in the memory section 307 of a record regenerator, for example, the internal memory in drawing 3 , and the storage element of others which the record regenerator 300 has, and key block assignment that it was accessed by the control section 301 of drawing 3 at the time of authentication processing, and setting information was followed is performed.

[0185] The master key MKake for storage device authentication keys of the internal memory 307 of the record regenerator 300 is a master key for authentication keys set up according to the setup of each assignment key block, the correspondence of it only in an assignment key block is attained, and the mutual recognition with key blocks other than an assignment key block has become with the configuration which is not materialized.

[0186] the key block of N individual of 1 - N is set to the internal memory 405 of a storage device 400, and storage device identification information, an authentication key, initial value, a preservation key, a random-number generation key, and a kind store for every key block so that I may be understood from drawing 19 -- having -- the hook for authentication at least -- data are stored as different data for every block.

[0187] Thus, the key data configurations of a key block of a storage device 400 differ for every block. The key block which can perform authentication processing using the master key MKake for storage device authentication keys with which it followed, for example, a certain record playback device A was stored in the internal memory is key block No.1, and it becomes that another key block of the key block which can attest

the record regenerator B of another specification, for example, set up like key block No.2, is possible.

[0188] Although the latter part explains to a detail further, in case contents are stored in the external memory 402 of a storage device 400, encryption processing will be made and stored using the preservation key Kstr stored in each key block. More specifically, encryption processing of the contents key which enciphers a contents block is carried out with a preservation key.

[0189] As shown in drawing 19 , the preservation key is constituted as a different key for every block. Therefore, using in common the contents stored in the memory of one certain storage device between two record regenerators of a different setup set up so that a different key block might be specified in both is prevented. That is, the record regenerator with which a different setup was made can use only the contents stored in the storage device corresponding to each setup.

[0190] In addition, the data which can be communalized about each key block may be constituted so that only the key data for authentication and preservation key data may be differed possible [also communalizing].

[0191] It sets up or there is an example set up so that the key block number which should be specified, for example according to model (a non-portable type, pocket mold, etc.) of record regenerator 300 may be differed as an example which constitutes the key block which becomes such a storage device from the key data with which plurality differs and an assignment key block may be differed for every application. Furthermore, the record regenerator which sets an assignment key block to No.1 about the record regenerator sold in Japan, for example, and is sold in the U.S. can also be considered as the configuration which performs a key block setup which is different for every area so that an assignment key block may be set to No.2. Since it will be impossible to use with the record regenerator with which a different key setup was made even if a storage device like a memory card is transmitted to the U.S. from Japan or Japan from the U.S., the contents stored with a preservation key which is used in each different selling area and is different in a storage device by considering as such a configuration can prevent inaccurate and disorderly circulation of the contents stored in memory. The condition that the contents key Kcon specifically enciphered with a different preservation key Kstr becomes available mutually in two countries can be eliminated.

[0192] Furthermore, also in which record regenerator 300, at least one key block to the key block 1 of the internal memory 405 of the storage device 400 shown in drawing 19 - N, for example, the key block of No.N, may be constituted as a key block

available in common.

[0193] For example, by storing in all devices the master key MKake for storage device authentication keys in which authentication with key block No.N is possible, the whole application, it can force and can treat independently for every country the model exception of record regenerator 300 as contents which can circulate. For example, the encryption contents stored in the memory card with the preservation key stored in key block No.N turn into available contents in all devices. For example, music data etc. can be enciphered with the preservation key of a key block available in common, it can memorize to a memory card, and decode regeneration of the data from a memory card can be enabled by the thing which stored the common master key MKake for storage device authentication keys for this memory card too and which is set to the voice playback device of a pocket mold etc., for example.

[0194] The example of use of the storage device which has two or more key blocks which can be set to the data processor of this invention is shown in drawing 21 . the record regenerator 2101 -- the record regenerator of the product for Japan -- it is -- No. of a key block of a storage device -- it has the master key with which the authentication processing between 1 and 4 is materialized. the record regenerator 2102 -- the record regenerator of the product for USs -- it is -- No. of a key block of a storage device -- it has the master key with which the authentication processing between 2 and 4 is materialized. the record regenerator 2103 -- the record regenerator of the product for EU -- it is -- No. of a key block of a storage device -- it has the master key with which the authentication processing between 3 and 4 is materialized.

[0195] For example, the contents to which the record regenerator 2101 performed cipher processing through the preservation key which authentication was materialized between storage device A, the key block 1 of 2104, or the key block 4, and was stored in each key block are stored in external memory. The contents to which the record regenerator 2102 performed cipher processing through the preservation key which authentication was materialized between storage device B, the key block 2 of 2105, or the key block 4, and was stored in each key block are stored in external memory. The contents to which the record regenerator 2103 performed cipher processing through the preservation key which authentication was materialized between storage device C, the key block 3 of 2106, or the key block 4, and was stored in each key block are stored in external memory. Here, when the record regenerator 2102 or the record regenerator 2103 is equipped with storage device A and 2104, since authentication between the record regenerator 2102, the record regenerator 2103, and the key block

1 is not materialized, use of the contents by which cipher processing was made with the preservation key of the key block 1 becomes impossible. On the other hand, since authentication between the record regenerator 2102, the record regenerator 2103, and the key block 4 is materialized, the contents by which cipher processing was made with the preservation key of the key block 4 become available.

[0196] As mentioned above, in the data processor of this invention, the key block which consists of a key set with which plurality differs in a storage device is constituted, and since it considered as the configuration which, on the other hand, stores in a record playback device the master key to a specific key block which can be attested, it becomes possible to set up a use limit of the contents according to various use modes.

[0197] In addition, it is also good to make into plurality, for example, 1-k, the key block which can be specified in one record playback device, and to make into plurality the key block which can be specified like p-q in other record regenerators also as a configuration which establishes two or more key blocks available in common possible.

[0198] (7) Explain the processing which downloads contents from the record regenerator 300 to the external memory of a storage device 400 in the download processing to a storage device from a record regenerator, next the data processor of this invention.

[0199] Drawing 22 is a flow chart explaining the procedure which downloads contents from the record regenerator 300 to a storage device 400. In addition, in drawing 22 , the mutual recognition processing already mentioned above between the record regenerator 300 and the storage device 400 shall be completed.

[0200] In step S51, the control section 301 of the record regenerator 300 reads the data according to a predetermined format from the media 500 which stored contents using the reading section 304, or receives data according to a predetermined format using the communications department 305 from means of communications 600. And the control section 301 of the record regenerator 300 transmits the header (Header) part of the data (refer to drawing 4) to the record regenerator cipher-processing section 302 of the record regenerator 300.

[0201] Next, the control section 306 of the record regenerator cipher-processing section 302 which received the header (Header) at step S51 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the check value A in step S52. The check value A is calculated according to the ICV count approach which used as the key the check value A generation key Kicva saved at the internal memory 307 of the record regenerator cipher-processing

section 302, and was explained by drawing 7 by making identification information (Content ID) and a handling plan (Usage Policy) into a message, as shown in drawing 23 . In addition, initial value saves the initial value IVa for check value A generation also as IV=0 at the internal memory 307 of the record regenerator cipher-processing section 302, and it may be used for it. Finally check value:ICVa stored in the check value A and the header (Header) is compared, and when in agreement, it progresses to step S53.

[0202] As previously explained in drawing 4 , the check values A and ICVa are check values for verifying the alteration of identification information and a handling plan. When in agreement with check value:ICVa by which the check value A calculated according to the ICV count approach which used as the key the check value A generation key Kicva saved at the internal memory 307 of the record regenerator cipher-processing section 302, and was explained by drawing 7 by making identification information (Content ID) and a handling plan (Usage Policy) into a message was stored in the header (Header), it is judged that there is no alteration of identification information and a handling plan.

[0203] Next, the control section 306 of the record regenerator cipher-processing section 302 makes the delivery key Kdis generate in step S53 in the code / decryption section 308 of the record regenerator cipher-processing section 302. As a generation method of the delivery key Kdis, it generates as follows, for example.

[0204]

[Equation 4]

$$Kdis=DES(MKdis, \text{Content ID} \hat{} IVdis)$$

[0205] MKdis is a master key for delivery keys for generating the delivery key Kdis here, and this is a key stored in the internal memory of the record regenerator 300 as mentioned above. Moreover, Content ID is the identification information of the header unit of contents data, and IVdis is the initial value for delivery keys further. Moreover, in the above-mentioned formula, DES() makes the 1st argument a cryptographic key, it is the function which enciphers the value of the 2nd argument, and operation ^ shows the exclusive OR of 64 bitwises.

[0206] In step S54 the control section 306 of the record regenerator cipher-processing section 302 The media 500 which received through the reading section 304 using the delivery key Kdis generated at step S53 using the code / decryption section 308 of the record regenerator cipher-processing section 302, Or decryption processing of the block information key Kbit and the contents key Kcon

(refer to drawing 4) stored in the header unit of data which received from means of communications 600 through the communications department 305 is performed. As shown in drawing 4 , on channels, such as media, such as DVD and CD, or the Internet, as for these block information key Kbit and the contents key Kcon, encryption processing is beforehand performed with the delivery key Kdis.

[0207] Furthermore, in step S55, the control section 306 of the record regenerator cipher-processing section 302 decrypts block information (BIT) with the block information key Kbit decrypted at step S54 using the code / decryption section 308 of the record regenerator cipher-processing section 302. As shown in drawing 4 , on channels, such as media, such as DVD and CD, or the Internet, as for block information (BIT), encryption processing is beforehand performed with the block information key Kbit.

[0208] Furthermore, in step S56, the control section 306 of the record regenerator cipher-processing section 302 divides the block information key Kbit, the contents key Kcon, and block information (BIT) per 8 bytes, and carries out the exclusive OR of all them (which operations, such as addition and subtraction, may be used). Next, the control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the check value B (ICVb). As shown in drawing 24 , the check value B uses as a key the check value B generation key Kicvb saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates the exclusive-OR value which calculated the point by DES. As compared with the last, ICVb within the check values B and Header is progressed to step S57, when in agreement.

[0209] As previously explained in drawing 4 , the check values B and ICVb are check values for verifying the alteration of the block information key Kbit, the contents key Kcon, and block information (BIT). The check value B generation key Kicvb saved at the internal memory 307 of the record regenerator cipher-processing section 302 is used as a key. The check value B which enciphered and generated the value acquired by dividing the block information key Kbit, the contents key Kcon, and block information (BIT) per 8 bytes, and carrying out an exclusive OR by DES When in agreement with check value:ICVb stored in the header (Header), it is judged that there is no alteration of the block information key Kbit, the contents key Kcon, and block information.

[0210] The control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator

cipher-processing section 302 calculate a middle check value in step S57. As shown in drawing 25 , a middle check value uses as a key the total check value generation key Kicvt saved at the internal memory 307 of the record regenerator cipher-processing section 302, and calculates it according to the ICV count approach explained by drawing 7 by making into a message the check value A, the check value B, and all the held contents check values in the verified header (Header). In addition, the total initial value IVt for check value generation is saved also as initial value IV=0 at the internal memory 307 of the record regenerator cipher-processing section 302, and it may be used. Moreover, the generated middle check value is held in the record regenerator cipher-processing section 302 of the record regenerator 300 if needed.

[0211] This middle check value is generated considering the check value A, the check value B, and all contents check values as a message, and may perform verification about the data set as the verification object of each of these check values by collating processing of a middle check value. In this example However, the un-altering nature verification processing as share data of the whole system, In order to distinguish the verification processing for identifying as occupancy data which only each record playback device 300 occupies and to make activation possible after download processing Based on the middle check value, generation of the check value ICVt from which plurality differs further, i.e., the total check value, and the record regenerator proper check value ICVdev is separately enabled from the middle check value. The latter part explains these check values.

[0212] The control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the total check value ICVt. As shown in drawing 25 , the total check value ICVt uses as a key the system signature key Ksys saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates a middle check value by DES. ICVt in Header saved at the generated total check value ICVt and step S51 at the end is compared, and when in agreement, it progresses to step S58. The system signature key Ksys is a signature key which is common in the whole system set which performs two or more record regenerators, i.e., record regeneration of a certain fixed data.

[0213] As previously explained in drawing 4 , the total check value ICVt is a check value for verifying the alteration of all the check values of ICVa, ICVb, and each contents block. Therefore, when in agreement with check value:ICVt by which the total check value generated by above-mentioned processing was stored in the header (Header), it is judged that there is no alteration of all the check values of ICVa, ICVb,

and each contents block.

[0214] Next, in step S58, the control section 301 of the record regenerator 300 takes out the contents block information within block information (BIT), and investigates whether there is any paddle with which the contents block is a candidate for verification. When the contents block is a candidate for verification, the contents check value is stored in the block information in a header.

[0215] When the contents block has become a candidate for verification, the corresponding contents block is read from media 500 using the reading section 304 of the record regenerator 300, or it receives from means of communications 600 using the communications department 305 of the record regenerator 300, and transmits to the record regenerator cipher-processing section 302 of the record regenerator 300. The control section 306 of the record regenerator cipher-processing section 302 which received this makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate a contents mean value.

[0216] A contents mean value decrypts the inputted contents block in the CBC mode of DES, and it divides every 8 bytes, and it is the contents key Kcon decrypted at step S54, and it generates [the exclusive OR of the result (which operations, such as addition and subtraction, may be used) is carried out altogether, and] it.

[0217] Next, the control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate a contents check value. A contents check value uses as a key the contents check value generation key Kicvc saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates a contents mean value by DES. And the control section 306 of the record regenerator cipher-processing section 302 compares with the contents check value concerned ICV within the contents block received from the control section 301 of the record regenerator 300 at step S51, and passes the result to the control section 301 of the record regenerator 300. When having succeeded in verification, the control section 301 of the record regenerator 300 which received this repeats the same verification processing until it takes out the following contents block for verification, and it makes the record regenerator cipher-processing section 302 of the record regenerator 300 verify it and it verifies all contents blocks. In addition, as long as it doubles the Header generation side, the initial value IVc for contents check value generation is saved also as IV=0 at the internal memory 307 of the record regenerator cipher-processing section 302, and it may be used. Moreover, all the checked contents check values are held in the record regenerator cipher-processing

section 302 of the record regenerator 300. When the record regenerator cipher-processing section 302 of the record regenerator 300 supervised the verification sequence of the contents block for verification, and sequence was wrong or it was made to verify the same contents block twice or more, it should fail in authentication further again. And when all verification is successful, it progresses to step S59.

[0218] Next, the record regenerator cipher-processing section 302 of the record regenerator 300 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 encipher the block information key Kbit decrypted at step S54, and the contents key Kcon in step S59 with the session key Kses shared on the occasion of mutual recognition. The control section 301 of the record regenerator 300 reads the block information key Kbit and the contents key Kcon which were enciphered with the session key Kses from the record regenerator cipher-processing section 302 of the record regenerator 300, and transmits these data to a storage device 400 through the record device controller 303 of the record regenerator 300.

[0219] Next, the storage device 400 which received the block information key Kbit transmitted from the record regenerator 300 and the contents key Kcon is made to decrypt with the session key Kses which shared the received data on the occasion of mutual recognition in the code / decryption section 406 of the storage device cipher-processing section 401, and is made to re-encipher in step S60 with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401. Finally, the control section 301 of the record regenerator 300 reads the block information key Kbit and the contents key Kcon which were re-enciphered with the preservation key Kstr from a storage device 400 through the record device controller 303 of the record regenerator 300. And these keys are transposed to the block information key Kbit and the contents key Kcon which were enciphered with the delivery key Kdis.

[0220] the contents to which the control section 301 of the record regenerator 300 took out and downloaded use limit information from the handling plan (Usage Policy) of the header unit of data in step S61 -- the record regenerator 300 concerned -- it can use (use limit information sets it as 1 in this case) -- same another record regenerator 300 -- it can use (use limit information sets it as 0 in this case) -- it judges. When use limit information is 1 as a result of a judgment, it progresses to step S62.

[0221] The control section 301 of the record regenerator 300 makes the record regenerator cipher-processing section 302 of the record regenerator 300 calculate

the check value of a record regenerator proper in step S62. The check value of a record regenerator proper uses as a key the record regenerator signature key Kdev saved at the internal memory 307 of the record regenerator cipher-processing section 302, as shown in drawing 25 , and it enciphers and generates the middle check value held at step S58 by DES. The check value ICVdev of the calculated record regenerator proper is overwritten instead of the total check value ICVt.

[0222] As explained previously, the system signature key Ksys is a system signature key used since a signature or ICV common to a distribution system is attached, and is a record regenerator signature key used since the record regenerator signature keys Kdev differ for every record regenerator and a record regenerator attaches a signature or ICV. Namely, although the data signed with the system signature key Ksys become available in common since the check of Success ICVt, i.e., the total check value, will correspond by the system (record regenerator) which has the same system signature key When signed using the record regenerator signature key Kdev Since a record regenerator signature key is a key of a proper at the record regenerator, the data stored in the storage device after the data signed using the record regenerator signature key Kdev, i.e., a signature When other record regenerators tend to be equipped with the storage device and it is going to reproduce, since the check value ICVdev of a record regenerator proper becomes inharmonious and serves as an error, it can reproduce.

[0223] Therefore, in the data processor of this invention, it becomes possible to set up the contents which can be used common to a system by setup of use limit information, and the contents which can be used for a record regenerator proper free.

[0224] In step S63, the control section 301 of the record regenerator 300 saves contents at the external memory 402 of a storage device 400.

[0225] Drawing 26 is drawing showing the contents situation in a storage device in case use limit information is 0. Drawing 27 is drawing showing the contents situation in a storage device in case use limit information is 1. The point that drawing 26 differs from drawing 4 is whether the contents block information key Kbit and the contents key Kcon are enciphered with the delivery key Kdis, or enciphered with the preservation key Kstr. Moreover, the point that drawing 27 differs from drawing 26 is enciphered with the record regenerator signature key Kdev of a record regenerator proper by drawing 27 to the check value calculated from a middle check value being enciphered with the system signature key Ksys by drawing 26 .

[0226] In addition, in the processing flow of drawing 22 , when verification of the check value A goes wrong at step S52, verification of the check value B goes wrong at step

S56, verification of the total check value ICVt goes wrong at step S57 and verification of the contents check value of each contents block goes wrong at step S58, it progresses to step S64 and a predetermined error message is performed.

[0227] Moreover, when use limit information is 0 at step S61, step S62 is skipped and it progresses to step S63.

[0228] (8) regeneration with the record regenerator of storage device storing information -- explain regeneration with the record regenerator 300 of the contents information stored in the external memory 402 of a storage device 400 next.

[0229] Drawing 28 is a flow chart explaining the procedure of the record regenerator 300 reading contents from a storage device 400, and using contents. In addition, also in drawing 28 , mutual recognition shall already be completed between the record regenerator 300 and a storage device 400.

[0230] In step S71, the control section 301 of the record regenerator 300 reads contents from the external memory 402 of a storage device 400 using the record device controller 303. And the control section 301 of the record regenerator 300 transmits the header (Header) part of the data to the record regenerator cipher-processing section 302 of the record regenerator 300. Step S72 is the same processing as step S52 explained in "download processing to a storage device from (7) record regenerator", and the control section 306 of the record regenerator cipher-processing section 302 which received the header (Header) is the processing which makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the check value A. The check value A uses as a key the check value A generation key Kicva saved at the internal memory 307 of the record regenerator cipher-processing section 302, as shown in drawing 23 explained previously, and it is calculated according to the same ICV count approach with drawing 7 having explained by making identification information (Content ID) and a handling plan (Usage Policy) into a message.

[0231] As explained previously, the check values A and ICVa are check values for verifying the alteration of identification information and a handling plan. The check value A generation key Kicva saved at the internal memory 307 of the record regenerator cipher-processing section 302 is used as a key. The check value A calculated according to the ICV count approach explained by drawing 7 by making identification information (Content ID) and a handling plan (Usage Policy) into a message When in agreement with check value:ICVa stored in the header (Header), it is judged that there is no alteration of the identification information stored in the storage device 400 and a handling plan.

[0232] Next, in step S73, the control section 301 of the record regenerator 300 takes out the block information key Kbit and the contents key Kcon from the read header (Header) part, and transmits to a storage device 400 through the record device controller 303 of the record regenerator 300. The code / decryption section 406 of the storage device cipher-processing section 401 are made to carry out decryption processing with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401, and the storage device 400 which received the block information key Kbit transmitted from the record regenerator 300 and the contents key Kcon makes it re-encipher the received data with the session key Kses shared on the occasion of mutual recognition. And the control section 301 of the record regenerator 300 reads the block information key Kbit and the contents key Kcon which were re-enciphered with the session key Kses from a storage device 400 through the record device controller 303 of the record regenerator 300.

[0233] Next, in step S74, the control section 301 of the record regenerator 300 transmits the block information key Kbit and the contents key Kcon which were re-enciphered with the received session key Kses to the record regenerator cipher-processing section 302 of the record regenerator 300.

[0234] The record regenerator cipher-processing section 302 of the record regenerator 300 which received the block information key Kbit re-enciphered with the session key Kses and the contents key Kcon makes the code / decryption section 308 of the record regenerator cipher-processing section 302 decrypt the block information key Kbit enciphered with the session key Kses, and the contents key Kcon with the session key Kses shared on the occasion of mutual recognition. And the block information received at step S71 with the decrypted block information key Kbit is made to decrypt.

[0235] In addition, the decrypted block information key Kbit, the contents key Kcon, and the block information BIT are transposed to the block information key Kbit received at step S71, the contents key Kcon, and the block information BIT, and the record regenerator cipher-processing section 302 of the record regenerator 300 holds them. Moreover, the control section 301 of the record regenerator 300 reads the decrypted block information BIT from the record regenerator cipher-processing section 302 of the record regenerator 300.

[0236] Step S75 is the same processing as step S56 explained in "download processing to a storage device from (7) record regenerator." The control section 306 of the record regenerator cipher-processing section 302 divides the block information

key Kbit read from the storage device 400, the contents key Kcon, and block information (BIT) per 8 bytes, and carries out the exclusive OR of all them. Next, the control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the check value B (ICVb). As shown in drawing 24 explained previously, the check value B uses as a key the check value B generation key Kicvb saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates the exclusive-OR value which calculated the point by DES. As compared with the last, ICVb within the check values B and Header is progressed to step S76, when in agreement.

[0237] As explained previously, the check values B and ICVb are check values for verifying the alteration of the block information key Kbit, the contents key Kcon, and block information. The check value B generation key Kicvb saved at the internal memory 307 of the record regenerator cipher-processing section 302 is used as a key. The check value B which enciphered and generated the value acquired by dividing the block information key Kbit read from the storage device 400, the contents key Kcon, and block information (BIT) per 8 bytes, and carrying out an exclusive OR by DES When in agreement with check value:ICVb stored in the header (Header) in the data read from the storage device 400, it is judged that there is no alteration of the block information key Kbit of the data stored in the storage device 400, the contents key Kcon, and block information.

[0238] The control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate a middle check value in step S76. A middle check value uses as a key the total check value generation key Kicvt saved at the internal memory 307 of the record regenerator cipher-processing section 302 as shown in drawing 25 explained previously, and calculates it according to the ICV count approach explained by drawing 7 etc. by making into a message the check value A, the check value B, and all the held contents check values in the verified header (Header). In addition, initial value saves IVt also as IV=0 at the total initial value for check value generation at the internal memory 307 of the record regenerator cipher-processing section 302, and may use it. Moreover, the generated middle check value is held in the record regenerator cipher-processing section 302 of the record regenerator 300 if needed.

[0239] next, the contents which the control section 301 of the record regenerator 300 took out use limit information from the handling plan (Usage Policy) contained in the

header unit of the data read from the external memory 402 of a storage device 400, and were downloaded in step S77 -- the record regenerator 300 concerned -- it can use (use limit information is 1) -- same another record regenerator 300 -- it can use (use limit information is 0) -- it judges. When the use limit which use limit information can use by 1 and the downloaded contents can use only with the record regenerator 300 concerned is set up as a result of the judgment, it progresses to step S80, and when it is a setup which use limit information can use by 0 [300], i.e., same another record regenerator, it progresses to step S78. In addition, the cipher-processing section 302 may perform processing of step S77.

[0240] In step S78, count of the same total check value ICVt as step S58 explained in the download processing to a storage device from (7) record regenerator is performed. That is, the control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the total check value ICVt. The total check value ICVt uses as a key the system signature key Ksys saved at the internal memory 307 of the record regenerator cipher-processing section 302, as shown in drawing 25 explained previously, and it enciphers and generates a middle check value by DES.

[0241] Next, it progresses to step S79, ICVt in the header (Header) saved at the total check value ICVt generated in step S78 and step S71 is compared, and when in agreement, it progresses to step S82.

[0242] As explained previously, the total check value ICVt is a check value for verifying the alteration of all the check values of ICVa, ICVb, and each contents block. Therefore, when in agreement with check value:ICVt by which the total check value generated by above-mentioned processing was stored in the header (Header), in the data stored in the storage device 400, it is judged that there is no alteration of all the check values of ICVa, ICVb, and each contents block.

[0243] In a judgment at step S77, when the downloaded contents are setup which can be used only with the record regenerator 300 concerned (i.e., when setting information is 1), it progresses to step S80.

[0244] The control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the check value ICVdev of a record regenerator proper in step S80. The check value ICVdev of a record regenerator proper uses as a key the record regenerator signature key Kdev of a record regenerator proper saved at the internal memory 307 of the record regenerator cipher-processing section 302, as shown in drawing 25 explained previously, and it

enciphers and generates a middle check value by DES. In step S81, ICVdev in Header saved at the check value ICVdev and step S71 of the record regenerator proper calculated at step S80 is compared, and when in agreement, it progresses to step S82.

[0245] Thus, the data signed with the system signature key Ksys A check is successful by the system (record regenerator) which has the same system signature key. namely, since the total check value ICVt will be in agreement, when it becomes available in common and is signed using the record regenerator signature key Kdev Since a record regenerator signature key is a key of a proper at the record regenerator, the data stored in the storage device after the data signed using the record regenerator signature key Kdev, i.e., a signature When other record regenerators tend to be equipped with the storage device and it is going to reproduce, since the check value ICVdev of a record regenerator proper becomes inharmonious and serves as an error, it can reproduce. Therefore, it becomes possible to set up the contents which can be used common to a system, and the contents which can be used for a record regenerator proper free by setup of use limit information.

[0246] In step S82, the control section 301 of the record regenerator 300 takes out the contents block information within the block information BIT read at step S74, and investigates whether there is any paddle with which the contents block is a candidate for encryption. When having become a candidate for encryption, the corresponding contents block is read from the external memory 402 of a storage device 400 through the record device controller 303 of the record regenerator 300, and it transmits to the record regenerator cipher-processing section 302 of the record regenerator 300. When the contents block is a candidate for verification, it makes a contents check value verify in the following step S83, while the control section 306 of the record regenerator cipher-processing section 302 which received this makes the code / decryption section 308 of the record regenerator cipher-processing section 302 decrypt contents.

[0247] Step S83 is the same processing as step S58 explained in "download processing to a storage device from (7) record regenerator." The control section 301 of the record regenerator 300 takes out the contents block information within block information (BIT), when it judges whether there is any paddle with which the contents block is a candidate for verification from the storing situation of a contents check value and the contents block has become a candidate for verification about it, receives the corresponding contents block from the external memory 402 of a storage device 400, and transmits to the record regenerator cipher-processing section 302 of the record regenerator 300. The control section 306 of the record regenerator

cipher-processing section 302 which received this makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate a contents mean value.

[0248] The inputted contents block is decrypted in the CBC mode of DES, a contents mean value divides into 8 bytes, and it is the contents key Kcon decrypted at step S74, and it generates [the exclusive OR of the result is carried out altogether, and] it.

[0249] Next, the control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate a contents check value. A contents check value uses as a key the contents check value generation key Kicvc saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates a contents mean value by DES. And the control section 306 of the record regenerator cipher-processing section 302 compares with the contents check value concerned ICV within the contents block received from the control section 301 of the record regenerator 300 at step S71, and passes the result to the control section 301 of the record regenerator 300. When having succeeded in verification, the control section 301 of the record regenerator 300 which received this repeats the same verification processing until it takes out the following contents block for verification, and it makes the record regenerator cipher-processing section 302 of the record regenerator 300 verify it and it verifies all contents blocks. In addition, initial value saves the initial value IVc for contents check value generation also as IV=0 at the internal memory 307 of the record regenerator cipher-processing section 302, and may use it. Moreover, all the checked contents check values are held in the record regenerator cipher-processing section 302 of the record regenerator 300. When the record regenerator cipher-processing section 302 of the record regenerator 300 supervised the verification sequence of the contents block for verification, and sequence was wrong or it was made to verify the same contents block twice or more, it should fail in authentication further again.

[0250] The control section 301 of the record regenerator 300 takes out the contents decrypted from the record regenerator cipher-processing section 302 of the record regenerator 300, when receiving the comparison result (all comparison results are considered as a success when it is not a candidate for verification) of the contents check value concerned and having succeeded in verification. And it repeats until take out the following contents block for a decryption, it makes the record regenerator cipher-processing section 302 of the record regenerator 300 decrypt and it decrypts all contents blocks.

[0251] In addition, in step S83, when it becomes an inequality in verification processing of a contents check value, the record regenerator cipher-processing section 302 of the record regenerator 300 stops processing at the time as a verification failure, and does not perform a decryption of the contents which remain. Moreover, when the record regenerator cipher-processing section 302 of the record regenerator 300 supervised the decryption sequence of the contents block for a decryption, and sequence was wrong or it was made to decrypt the same contents block twice or more, it should fail in the decryption.

[0252] In addition, when verification of the check value A goes wrong at step S72 and verification of the check value B goes wrong at step S75, When verification of the total check value ICVt goes wrong at step S79 and verification of the check value ICVdev of a record regenerator proper goes wrong at step S81 When verification of the contents check value of each contents block goes wrong at step S83, it progresses to step S84 and a predetermined error message is performed.

[0253] As explained above, in case contents are downloaded or used Encipher important data and contents, and do not concealment-ize or alteration verification does not come out as much as possible. Since the contents key Kcon for decrypting the block information key Kbit and contents for decrypting the block information BIT is saved with the preservation key Kstr of a storage device proper, Even if it reproduces the data on an archive medium to another archive medium simply, contents cannot be decrypted correctly and can be carried out. It is because it has the configuration which cannot decrypt data correctly in another storage device in order to decrypt the data more specifically enciphered with a different preservation key Kstr for every storage device in step S74 of drawing 28 .

[0254] (9) There is a point of having enabled use of a storage device after the mutual recognition processing performed between the record regenerators 300 and storage devices 400 which were mentioned above, and having restricted the use mode to one of the descriptions in the data processor of key message-exchange this invention after mutual recognition.

[0255] For example, in order to eliminate generating storage devices, such as a memory card which stored contents, setting this to a record regenerator, and being used by the unjust duplicate etc. The transfer between the record regenerator 300 of contents (enciphered) and a storage device 400 is enabled the condition [having performed mutual recognition processing between the record regenerator 300 and a storage device 400, and having become Authentication O.K.].

[0256] In order to realize the above-mentioned restrictive processing, in the data

processor of this invention, all processings in the cipher-processing section 401 of a storage device 400 have composition performed based on the series of commands set up beforehand. That is, a storage device has the command-processing configuration which takes out the command based on a command number from a register one by one, and executes it. Drawing explaining the command-processing configuration in this storage device is shown in drawing 29 .

[0257] A command number (No.) is outputted from the record device controller 303 to the communications department (a receive register is included) 404 of a storage device 400 at the basis of control of the control section 301 of the record regenerator 300 between the record regenerator 300 which has the record regenerator cipher-processing section 302 as shown in drawing 29 , and the storage device 400 which has the storage device cipher-processing section 401.

[0258] A storage device 400 has the command number Management Department 2201 in the control section 403 in the cipher-processing section 401. The command number Management Department 2901 holds the command register 2902, and stores the series of commands corresponding to the command number outputted from the record regenerator 300. As series of commands is shown in the right of drawing 29 , the execute command is matched to the command number one by one from the command number 0 to y. The command number Management Department 2901 supervises the command number outputted from the record regenerator 300, takes out a corresponding command from a command register 2902, and executes it.

[0259] The command sequence stored in the command register 2902 is matched with the command number 0 which the series of commands about an authentication processing sequence precedes - k as shown in the right of drawing 29 . Furthermore, decode, key exchange, and the cipher-processing command sequence 2 are matched with command number p-s after the series of commands about an authentication processing sequence by decode, key exchange, the cipher-processing command sequence 1, and command number u-y that follows further.

[0260] If the record regenerator 300 is equipped with a storage device 400 as previously explained in the authentication processing flow of drawing 20 , the control section 301 of the record regenerator 300 will transmit an initialization instruction to a storage device 400 through the record device controller 303. In the control section 403 of the storage device cipher-processing section 401, the storage device 400 which received this receives an instruction through the communications department 404, and clears the authentication flag 2903. That is, it is set as the condition of not attesting. Or when a power source is supplied to a storage device 400 from the record

regenerator 300, the method which sets as a condition of not recognizing, at the time of power-on may be used.

[0261] Next, the control section 301 of the record regenerator 300 transmits an initialization instruction to the record regenerator cipher-processing section 302. At this time, a storage device insertion opening number is also transmitted collectively. By transmitting a storage device insertion opening number, even if it is the case where two or more storage devices are connected to the record regenerator 300, authentication processing with two or more storage devices 400 and data transmission and reception are attained at coincidence.

[0262] The record regenerator cipher-processing section 302 of the record regenerator 300 which received the initialization instruction clears the authentication flag 2904 corresponding to a storage device insertion opening number in the control section of the record regenerator cipher-processing section 302. That is, it is set as the condition of not attesting.

[0263] If these initialization processings are completed, the control section 301 of the record regenerator 300 will output a command number to ascending order one by one from the command number 0 through the record device controller 303. The command number Management Department 2901 of a storage device 400 supervises the command number inputted from the record regenerator 300, checks that a sequential input is carried out from 0, takes out a corresponding command from a command register 2902, and performs various processings, such as authentication processing. it considers as an error and possible, an initial state, i.e., activation, of a command number reception value, when the command number inputted is not regular order -- it resets to command number =0.

[0264] The command number is given so that the command sequence stored in the command register 2902 as shown in drawing 29 may precede and process authentication processing, and the processing sequence of decode, key exchange, and encryption processing is stored in subsequent processing.

[0265] The example of the processing sequence of decode, key exchange, and encryption processing is explained using drawing 30 and 31.

[0266] Drawing 30 constitutes a part of processing performed in download processing of the contents from the record regenerator 300 previously explained in drawing 22 to a storage device 400. It is [0267] specifically performed among steps S59-S60 in drawing 22 . In drawing 30 , step S3001 is processing whose storage device receives the data (ex. block information key Kbit, the contents key Kcon) enciphered with the session key Kses from the record regenerator, and series-of-commands p-s shown

by above-mentioned drawing 29 is started after that. The authentication processing command 0 - k are completed, and series-of-commands p-s is started after a flag [finishing / authentication] is set to the authentication flags 2903 and 2904 shown in drawing 29 . This is guaranteed when the command number Management Department 2901 receives a command number only in ascending order from 0.

[0268] Step S3002 is processing which stores in a register the data (ex. block information key Kbit, the contents key Kcon) as which the storage device was enciphered with the session key Kses received from the record regenerator.

[0269] Step S3003 is a step which performs processing which picks out from a register the data (ex. block information key Kbit, the contents key Kcon) enciphered with the session key Kses, and is decoded with the session key Kses.

[0270] Step S3004 is a step which performs processing which enciphers the data (ex. block information key Kbit, the contents key Kcon) decrypted with the session key Kses with the preservation key Kstr.

[0271] The above-mentioned processing steps 3002-3004 are processings included in command number p-s in the command register explained by previous drawing 29 . According to command number p-s which receives from the record regenerator 300 at the command number Management Department 2901 of a storage device 400, the storage device cipher-processing section 401 carries out sequential execution of these processings.

[0272] The following step S3005 is a step which stores in the external memory of a storage device the data (ex. block information key Kbit, the contents key Kcon) enciphered with the preservation key Kstr. In this step, the data which the record regenerator 300 enciphered with the preservation key Kstr may be read from the storage device cipher-processing section 401, and you may store in the external memory 402 of a storage device 400 after that.

[0273] The above-mentioned steps S3002-S3004 are activation sequences which are performed continuously and which cannot be interrupted, for example, it is at the decode processing termination time of step S3003, and since the read-out command differs from the command number of the ascending order set as command number p-s of a command register 2902 even if there is a data read-out instruction from the record regenerator 300, the command number Management Department 2901 does not receive activation of read-out. Therefore, it becomes impossible to read the decode data generated in the case of the key exchange in a storage device 400 from the exterior 300, for example, a record regenerator, and it can prevent unjust read-out of key data and contents.

[0274] Drawing 31 constitutes a part of processing performed in the contents regeneration which reads contents from the storage device 400 previously explained in drawing 28 , and is reproduced in the record regenerator 300. It is the processing specifically performed in step S73 in drawing 28 .

[0275] In drawing 31 , step S3101 is a step which performs read-out of the data (ex. block information key Kbit, the contents key Kcon) enciphered with the preservation key Kstr from the external memory 402 of a storage device 400.

[0276] Step S3102 is a step which stores in a register the data (ex. block information key Kbit, the contents key Kcon) enciphered with the preservation key Kstr read from the memory of a storage device. In this step, the data which the record regenerator 300 enciphered with the preservation key Kstr may be read from the external memory 402 of a storage device 400, and you may store in the register of a storage device 400 after that.

[0277] Step S3103 is a step which picks out from a register the data (ex. block information key Kbit, the contents key Kcon) enciphered with the preservation key Kstr, and carries out decode processing with the preservation key Kstr.

[0278] Step S3104 is a step which carries out encryption processing of the data (ex. block information key Kbit, the contents key Kcon) decrypted with the preservation key Kstr with the session key Kses.

[0279] The above-mentioned processing steps 3102-3104 are processings included in command number u-y in the command register explained by previous drawing 29 . According to command number u-y which receives from the record regenerator 300 at the command number Management Department 2901 of a storage device, the storage device cipher-processing section 406 carries out sequential execution of these processings.

[0280] The following step S3105 is processing which transmits the data (ex. block information key Kbit, the contents key Kcon) enciphered with the session key Kses to a record regenerator from a storage device.

[0281] The above-mentioned steps S3102-S3104 are activation sequences which are performed continuously and which cannot be interrupted, for example, it is at the decode processing termination time of step S3103, and since the read-out command differs from the command number of the ascending order set as command number u-y of a command register 2902 even if there is a data read-out instruction from the record regenerator 300, the command number Management Department 2901 does not receive activation of read-out. Therefore, it becomes impossible to read the decode data generated in the case of the key exchange in a storage device 400 from

the exterior 300, for example, a record regenerator, and it can prevent unjust read-out of key data or contents.

[0282] In addition, although the object decoded and enciphered by key exchange showed the example which are the block information key Kbit and the contents key Kcon by drawing 30 and the processing shown in 31, in the command sequence stored in the command register 2902 shown in these drawing 29 , the decode and the encryption processing accompanied by key exchange of the contents itself may include, and the object decoded and enciphered by key exchange is not limited to an above-mentioned example.

[0283] In the above, the key message exchange after the mutual recognition in the data processor of this invention was explained. Thus, since activation of the key message exchange in the data processor of this invention is attained and it has further composition which can prevent access from the outside of the decode data in the key message exchange after the authentication processing between a record regenerator and a storage device is completed, the advanced security of contents and key data is secured.

[0284] (10) The case where two or more contents data formats and the data format in the media 500 or means of communications 600 shown, for example in drawing 3 in the example corresponding to each format which downloaded and mentioned [regeneration] above are one class shown in drawing 4 has been explained. However, when not only the format shown in above-mentioned drawing 4 but contents are music, it is image data and it is programs, such as a game, as for media 500 or the data format in means of communications 600, it is desirable to adopt the data format according to contents. Hereafter, the download processing to the storage device corresponding to the data data format from which plurality differs, and each format, and the regeneration from a storage device are explained.

[0285] Four different data formats are shown in drawing 32 -35. The data format in case the data format on the media 500 shown in drawing 3 or means of communications 600 is stored by the external memory 402 of a storage device 400 on the right-hand side of each drawing again is shown in the left-hand side of each drawing. The outline of the data format shown in drawing 32 -35 is explained previously, and the difference between the contents of each data in each format and the data in each format is explained after that.

[0286] Drawing 32 is the format type 0 and is as common as the type which the **** explained and was shown as an example. This format type 0 of the description is the point which divide the whole data into the data block 1, i.e., a block, - Block N of the

magnitude of arbitration of N individual, encipher to arbitration about each block, and an encryption block and a non-enciphering block, i.e., a plaintext block, are made intermingled, and can constitute data. Encryption of a block is performed with the contents key Kcon, on media, it is enciphered with the delivery key Kdis and the contents key Kcon is enciphered with the preservation key Kstr stored in the internal memory of a storage device at the time of the preservation in a storage device. Also about the block information key Kbit, on media, it is enciphered with the delivery key Kdis and enciphered with the preservation key Kstr stored in the internal memory of a storage device at the time of the preservation in a storage device. These key exchange is performed according to the processing explained in the above-mentioned "key message exchange after (9) mutual recognition."

[0287] Drawing 33 is the format type 1, and although this format type 1 is dividing the whole data into the data block 1, i.e., a block, - Block N of N individual like the format type 0, it differs from the above-mentioned format type 0 at the point which made each block size of N individual the same magnitude. The encryption processing mode of the block with the contents key Kcon is the same as that of the above-mentioned format type 0. Moreover, the contents key Kcon which is enciphered with the delivery key Kdis on media, and is enciphered with the preservation key Kstr stored in the internal memory of a storage device at the time of the preservation in a storage device, and a block information key Kbit configuration are the same as that of the above-mentioned format type 0. The format type 1 is having considered as the fixed block configuration unlike the format type 0, and since configuration data, such as a data length for every block, are simplified, it becomes possible [reducing the memory size of block information as compared with the format type 0].

[0288] 1 set of encryption PERT and the PERT non-enciphering (plaintext) constitutes each block from the example of a configuration of drawing 33 . Thus, if the die length of a block and a configuration are regular, since it becomes unnecessary to check each block length and a block configuration in the cases, such as decode processing, efficient decode and cipher processing will become possible. In addition, in format 1, the PERT who constitutes each block, i.e., encryption PERT, and the PERT non-enciphering (plaintext) have composition which can be defined as a candidate for a check for every PERT, and when it is the block containing important point check parts, the contents check value ICVi is defined about the block.

[0289] Drawing 34 is the format type 2 and is this format type 2 of the description's being divided into the data block 1, i.e., a block, - Block N of the same magnitude of N individual, and enciphered with the block key Kblc according to individual about each

block, respectively. Encryption of each block key Kblc is performed with the contents key Kcon, on media, it is enciphered with the delivery key Kdis and the contents key Kcon is enciphered with the preservation key Kstr stored in the internal memory of a storage device at the time of the preservation in a storage device. Also about the block information key Kbit, on media, it is enciphered with the delivery key Kdis and enciphered with the preservation key Kstr stored in the internal memory of a storage device at the time of the preservation in a storage device.

[0290] Drawing 35 is the format type 3. This format type 3 of the description Like the format type 2, it is divided into the data block 1, i.e., a block, – Block N of the same magnitude of N individual, and is enciphered with the block key Kblc according to individual about each block, respectively, Furthermore, it is the point which encryption of each block key Kblc is enciphered with the delivery key Kdis on media not using a contents key, and is enciphered with the preservation key Kstr on the storage device. The contents key Kcon exists in neither on media and a device. On media, it is enciphered with the delivery key Kdis and the block information key Kbit is enciphered with the preservation key Kstr stored in the internal memory of a storage device at the time of the preservation in a storage device.

[0291] Next, the contents of data above-mentioned format type 0-3 are explained. As data were explained previously, it is roughly classified into two at a header unit and the contents section, and a contents identifier, a handling plan, the check values A and B, the total check value, a block information key, a contents key, and block information are included in a header unit.

[0292] For a handling plan, the data length of contents, header length, a format type (formats 0-3 explained below), For example, as the column of whether it is a program or it is data, and a contents type, the download to the storage device of the above-mentioned contents and playback explained The localization flag which is a flag which determines whether contents are available to a record regenerator proper, Furthermore, the authorization flag about a copy and MUBU processing, various kinds of use limit information concerning contents, such as contents encryption algorithm and the mode, further, and processing information on contents are stored.

[0293] Check value A:ICVa is a check value over identification information and a handling plan, for example, is generated by the technique explained by above-mentioned drawing 23 .

[0294] The block information key Kbit is a key for enciphering block information, and as explained previously, on media, it is enciphered with the delivery key Kdis and it is enciphered with the preservation key Kstr stored in the internal memory of a storage

device at the time of the preservation in a storage device.

[0295] The contents key Kcon is a key used for encryption of contents, and like the block information key Kbit, on media, it is enciphered with the delivery key Kdis and it is enciphered with the preservation key Kstr stored in the internal memory of a storage device by the format types 0 and 1 at the time of the preservation in a storage device. In addition, the contents key Kcon is used also for encryption of the block key Kblc constituted by contents each block by the format type 2. Moreover, the contents key Kcon does not exist in the format type 3.

[0296] Block information is a table which describes the information on each block, and the information which shows whether the magnitude of a block and the flag about whether it is enciphered or not, i.e., each block, are set as the object (ICV) of a check is stored. When the block is set as the object of a check, the check value ICVi of a block (check value of Block i) is defined and stored in a table. This block information is enciphered by the block information cryptographic key Kbit.

[0297] In addition, when the block is enciphered, the check value ICVi of a block, i.e., a contents check value, is generated as a value which enciphered the value which carried out exclusive OR of the whole plaintext (decode sentence) per 8 bytes with the contents check value generation key Kicvc in which it was stored by the internal memory 307 of the record regenerator 300. Moreover, when the block is not enciphered, it is generated as a value which inputted the whole block data (plaintext) into the alteration check value generating function (let DES-CBC-MAC and the contents check value generation key Kicvc be keys) shown in drawing 36 per 8 bytes, and obtained it. The example of a configuration which generates the check value ICVi of a contents block to drawing 36 is shown. Each of Message M constitutes 8 bytes each of decode sentence data or plaintext data.

[0298] In addition, in the format type 1, when at least one of the parts within a block is, the object data, i.e., the important point check parts, of the check value ICVi, the contents check value ICVi is defined about the block. Check value P-ICVij of the parts j in Block i is generated as a value which enciphered the value which carried out exclusive OR of the whole plaintext (decode sentence) per 8 bytes with the contents check value generation key Kicvc, when Parts j are enciphered. Moreover, when Parts j are not enciphered, it is generated as a value which inputted the whole data (plaintext) of a block of parts into the alteration check value generating function (let DES-CBC-MAC and the contents check value generation key Kicvc be keys) shown in drawing 36 per 8 bytes, and obtained it.

[0299] Furthermore, the parts which are [ICV flag =subject of ICV] which shows that

it is a candidate for a check in one block i, Namely, when one important point check parts exist Check value P-ICVij generated by above-mentioned technique is made into the check value ICVi of a block as it is. Moreover, when two or more parts which are [ICV flag =subject of ICV] which shows that it is a candidate for a check exist in one block i It is generated as a value which inputted two or more parts check value P-ICVij into the alteration check value generating function (let DES-CBC-MAC and the contents check value generation key Kicvc be keys) shown in drawing 37 per 8 bytes for the data connected with the parts numerical order, and obtained it. The example of a configuration which generates the contents check value ICVi of a contents block to drawing 37 is shown.

[0300] In addition, the check value ICVi of a block is not defined in the format types 2 and 3.

[0301] Check value B:ICVb is a check value over a block information key, a contents key, and the whole block information, for example, is generated by the technique explained by above-mentioned drawing 24 .

[0302] The total check value ICVt is a check value over above-mentioned check value A:ICVa, check value B:ICVb, and the whole check value ICVi contained in each block further set as the check object of contents, and is generated by performing encryption processing with the application of the system signature key Ksys to the middle check value generated from each check value, such as check value A:ICVa, as above-mentioned drawing 25 explained.

[0303] In addition, in the format types 2 and 3, the total check value ICVt is generated by performing encryption processing with the application of the system signature key Ksys to the middle check value generated from the data which connected the whole contents data, i.e., the contents data from the block key of block 1 to the last block, with above-mentioned check value A:ICVa and check value B:ICVb. The example of a configuration which generates the total check value ICVt in the format types 2 and 3 to drawing 38 is shown.

[0304] When, as for the proper check value ICVdev, the above-mentioned localization flag is set to 1, Namely, when it is shown that contents are available to a record regenerator proper It is the check value transposed to the total check value ICVt. In a format type 0 and 1 case It is generated as a check value over above-mentioned check value A:ICVa, check value B:ICVb, and the whole check value ICVi contained in each block further set as the check object of contents. It is generated by performing encryption processing with the application of the record regenerator signature key Kdev to the middle check value specifically generated from each check value, such as

check value A:ICVa, as above-mentioned drawing 25 or drawing 38 explained.

[0305] Next, download processing of the contents to a storage device 400 and the regeneration from the storage device 400 in the record regenerator 300 are explained using the flow of drawing 39 -44 from the record regenerator 300 in zero to format type 3 each.

[0306] First, download processing of the contents in the format types 0 and 1 is explained using drawing 39 .

[0307] The processing shown in drawing 39 is started by equipping with a storage device 400 the record regenerator 300 shown in drawing 3 . Step S101 is an authentication processing step between a record regenerator and a storage device, and is performed according to the authentication processing flow of drawing 20 explained previously.

[0308] When authentication processing of step S101 is completed and an authentication flag is set, the record regenerator 300 In step S102, from the media 500 which stored contents data [whether the data which followed the predetermined format through the reading section 304 are read, and] According to a predetermined format, data are received using the communications department 305 from means of communications 600, and the control section 301 of the record regenerator 300 transmits the header (Header) part of the data to the record regenerator cipher-processing section 302 of the record regenerator 300.

[0309] Next, the control section 306 of the cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the check value A in step S103. The check value A is calculated according to the ICV count approach which used as the key the check value A generation key Kicva saved at the internal memory 307 of the record regenerator cipher-processing section 302, and was explained using drawing 7 by making identification information (Content ID) and a handling plan (Usage Policy) into a message, as shown in drawing 23 . Next, in step S104, check value:ICVa stored in the check value A and the header (Header) is compared, and when in agreement, it progresses to step S105.

[0310] As explained previously, the check values A and ICVa are check values for verifying the alteration of identification information and a handling plan. When in agreement with check value:ICVa by which the check value A which uses as a key the check value A generation key Kicva saved at the internal memory 307 of the record regenerator cipher-processing section 302, and is calculated according to the ICV count approach by making identification information (Content ID) and a handling plan

(Usage Policy) into a message was stored in the header (Header), it is judged that there is no alteration of identification information and a handling plan.

[0311] Next, the control section 306 of the record regenerator cipher-processing section 302 makes the ejection of the delivery key Kdis, or generation perform in the code / decryption section 308 of the record regenerator cipher-processing section 302 in step S105. The generation method of the delivery key Kdis is performed like step S53 of drawing 22 explained previously using the master key MKdis for delivery keys.

[0312] Next, in step S106, decryption processing of the block information key Kbit and the contents key Kcon stored in the header unit of the media 500 which the control section 306 of the record regenerator cipher-processing section 302 received through the reading section 304 using the generated delivery key Kdis using the code / decryption section 308 of the record regenerator cipher-processing section 302, or the data received from means of communications 600 through the communications department 305 is performed.

[0313] Furthermore, in step S107, the control section 306 of the record regenerator cipher-processing section 302 decrypts block information with the decrypted block information key Kbit in the code / decryption section 308 of the record regenerator cipher-processing section 302.

[0314] Furthermore, in step S108, the control section 306 of the record regenerator cipher-processing section 302 generates the check value B (ICVb') from the block information key Kbit, the contents key Kcon, and block information (BIT). As shown in drawing 24 , the check value B uses as a key the check value B generation key Kicvb saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates the exclusive-OR value which consists of the block information key Kbit, a contents key Kcon, and block information (BIT) by DES. Next, in step S109, ICVb in the check value B and a header (Header) is compared, and when in agreement, it progresses to step S110.

[0315] As explained previously, the check values B and ICVb are check values for verifying the alteration of the block information key Kbit, the contents key Kcon, and block information. The check value B generation key Kicvb saved at the internal memory 307 of the record regenerator cipher-processing section 302 is used as a key. The check value B which enciphered and generated the value acquired by dividing the block information key Kbit, the contents key Kcon, and block information (BIT) per 8 bytes, and carrying out an exclusive OR by DES When in agreement with check value:ICVb stored in the header (Header), it is judged that there is no alteration of the

block information key Kbit, the contents key Kcon, and block information.

[0316] The control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate a middle check value in step S110. As shown in drawing 25 , a middle check value uses as a key the total check value generation key Kicvt saved at the internal memory 307 of the record regenerator cipher-processing section 302, and calculates it according to the ICV count approach explained by drawing 7 etc. by making into a message the check value A, the check value B, and all the held contents check values in verified Header. In addition, the generated middle check value is held in the record regenerator cipher-processing section 302 of the record regenerator 300 if needed.

[0317] Next, the control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate total check value ICVt' in step S111. As shown in drawing 25 , total check value ICVt' uses as a key the system signature key Ksys saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates a middle check value by DES. Next, in step S112, ICVt in generated total check value ICVt' and a header (Header) is compared, and when in agreement, it progresses to step S113.

[0318] As previously explained in drawing 4 , the total check value ICVt is a check value for verifying the alteration of all the check values of ICVa, ICVb, and each contents block. Therefore, when in agreement with check value:ICVt by which the total check value generated by above-mentioned processing was stored in the header (Header), it is judged that there is no alteration of all the check values of ICVa, ICVb, and each contents block.

[0319] Next, in step S113, the control section 301 of the record regenerator 300 takes out the contents block information within block information (BIT), and investigates whether there is any paddle with which the contents block is a candidate for verification. When the contents block is a candidate for verification, the contents check value is stored in the block information in a header.

[0320] When the contents block has become a candidate for verification, in step S114, the corresponding contents block is read from media 500 using the reading section 304 of the record regenerator 300, or it receives from means of communications 600 using the communications department 305 of the record regenerator 300, and transmits to the record regenerator cipher-processing section 302 of the record regenerator 300. The control section 306 of the record regenerator cipher-processing

section 302 which received this makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate contents check value ICVi'.

[0321] When the block is enciphered as having explained previously, contents check value ICVi' is the contents key Kcon, decrypts the inputted contents block in the CBC mode of DES, and enciphers and generates the contents mean value which carried out the exclusive OR of the whole of the result per 8 bytes, and generated it with the contents check value generation key Kicvc in which it was stored by the internal memory 307 of the record regenerator 300. Moreover, when the block is not enciphered, it is generated as a value which inputted the whole data (plaintext) into the alteration check value generating function (let DES-CBC-MAC and the contents check value generation key Kicvc be keys) shown in drawing 36 per 8 bytes, and obtained it.

[0322] Next, in step S115, the control section 306 of the record regenerator cipher-processing section 302 compares with the contents check value concerned ICV within the contents block received from the control section 301 of the record regenerator 300 at step S102, and passes the result to the control section 301 of the record regenerator 300. When having succeeded in verification, the control section 301 of the record regenerator 300 which received this repeats the same verification processing until it takes out the following contents block for verification, makes the record regenerator cipher-processing section 302 of the record regenerator 300 verify and verifies all contents blocks (step S116).

[0323] In addition, in step S104, step S109, step S112, or step S115, when coincidence of a check value is not obtained, download processing is ended as an error.

[0324] Next, the record regenerator cipher-processing section 302 of the record regenerator 300 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 encipher the block information key Kbit decrypted at step S106, and the contents key Kcon in step S117 with the session key Kses shared on the occasion of mutual recognition. The control section 301 of the record regenerator 300 reads the block information key Kbit and the contents key Kcon which were enciphered with the session key Kses from the record regenerator cipher-processing section 302 of the record regenerator 300, and transmits these data to a storage device 400 through the record device controller 303 of the record regenerator 300.

[0325] Next, the storage device 400 which received the block information key Kbit

transmitted from the record regenerator 300, and the contents key Kcon in step S118 The received data in the code / decryption section 406 of the storage device cipher-processing section 401 It is made to decrypt with the session key Kses shared on the occasion of mutual recognition. It is made to encipher again with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401. The control section 301 of the record regenerator 300 The block information key Kbit and the contents key Kcon which were re-enciphered with the preservation key Kstr are read from a storage device 400 through the record device controller 303 of the record regenerator 300. That is, the key of the block information key Kbit enciphered with the delivery key Kdis and the contents key Kcon chips, and **** is performed.

[0326] Next, in step S119, the control section 301 of the record regenerator 300 judges whether the contents which took out and downloaded use limit information from the handling plan (Usage Policy) of the header unit of data can use only with the record regenerator 300 concerned. It is shown that the downloaded contents can use this judgment also with same another record regenerator 300 when the contents downloaded when set as localization flag (use limit information) =1 can use only with the record regenerator 300 concerned and are set as localization flag (use limit information) =0. When it is localization flag (use limit information) =1 as a result of a judgment, it progresses to step S120.

[0327] The control section 301 of the record regenerator 300 makes the record regenerator cipher-processing section 302 of the record regenerator 300 calculate the check value of a record regenerator proper in step S120. The check value of a record regenerator proper uses the record regenerator signature key Kdev of a proper as a key at the record regenerator saved at the internal memory 307 of the record regenerator cipher-processing section 302, as shown in drawing 25 , and it enciphers and generates the middle check value generated at step S110 by DES. The check value ICVdev of the calculated record regenerator proper is overwritten instead of the total check value ICVt.

[0328] As explained previously, the system signature key Ksys is a system signature key used since a signature or ICV common to a distribution system is attached, and is a record regenerator signature key used since the record regenerator signature keys Kdev differ for every record regenerator and a record regenerator attaches a signature or ICV. Namely, although the data signed with the system signature key Ksys become available in common since the check of Success ICVt, i.e., the total check value, will correspond by the system (record regenerator) which has the same

system signature key When signed using the record regenerator signature key Kdev Since a record regenerator signature key is a key of a proper at the record regenerator, the data stored in the storage device after the data signed using the record regenerator signature key Kdev, i.e., a signature When other record regenerators tend to be equipped with the storage device and it is going to reproduce, since the check value ICVdev of a record regenerator proper becomes inharmonious and serves as an error, it can reproduce. In the data processor of this invention, the contents which can be used common to a system by setup of use limit information, and the contents which can be used for a record regenerator proper can be set up free.

[0329] Next, the control section 301 of the record regenerator 300 makes the record regenerator cipher-processing section 302 perform formation of a storing data format in step S121. As explained previously, a format type has each type to 0-3, is set up into the handling plan in a header (refer to drawing 5), and forms data according to the storing format on the right-hand side of drawing 32 -35 explained previously according to this setting type. Since the flow shown in this drawing 39 is either of the formats 0 and 1, it is formed in a format of drawing 32 and either of 33.

[0330] After formation of a storing data format is completed in step S121, in step 122, the control section 301 of the record regenerator 300 saves contents at the external memory 402 of a storage device 400.

[0331] The above is the mode of download processing of the contents data in the format types 0 and 1.

[0332] Next, download processing of the contents data in the format type 2 is explained using drawing 40 . It explains focusing on a different point from download processing of the above-mentioned format types 0 and 1.

[0333] Since steps S101-S109 are the same as that of download processing of the above-mentioned format types 0 and 1, explanation is omitted.

[0334] Since the contents check value ICVi is not defined as having explained previously, the format type 2 does not have the contents check value ICVi in block information. The middle check value in the format type 2 is generated by performing encryption processing with the application of the system signature key Ksys to the middle check value generated based on the data which connected the whole contents data from block [1st] initial data (block key of block 1) to the last block with the check value A and the check value B as shown in drawing 38 .

[0335] Therefore, in download processing of the format type 2, contents data are read in step S151, and generation of a middle check value is performed in step S152 based

on the check value A, the check value B, and the read contents data. In addition, contents data do not perform decode processing, even when enciphered.

[0336] By the format type 2, since decode of block data and inquiry processing of a contents check value are not performed like processing with the above-mentioned format types 0 and 1, quick processing is attained.

[0337] Step S Since 111 or less processing is the same as the processing in the format types 0 and 1, explanation is omitted.

[0338] The above is the mode of download processing of the contents data in the format type 2. As mentioned above, since decode of block data and inquiry processing of a contents check value are not performed like processing, download processing of the format type 2 is the format suitable for format type 0 and 1 data processing as which quick processing is attained and real-time operations, such as music data, are required.

[0339] Next, download processing of the contents data in the format type 3 is explained using drawing 41 . It explains focusing on a different point from download processing of the above-mentioned format types 0, 1, and 2.

[0340] Since steps S101-S105 are the same as that of download processing of the above-mentioned format types 0, 1, and 2, explanation is omitted.

[0341] Although the format type 3 has many parts which are fundamentally common in the processing in the format type 2, the format type 3 does not have a contents key, and it differs from the format type 2 in that the block key Kblc is enciphered and stored with the preservation key Kstr in a storage device.

[0342] The point which is different from the format type 2 in download processing of the format type 3 is explained as a core. By the format type 3, a block information key is decoded in step S161 which is degree step of step S105. Decryption processing of the block information key Kbit stored in the header unit of the media 500 which the control section 306 of the record regenerator cipher-processing section 302 received through the reading section 304 using the delivery key Kdis generated at step S105 using the code / decryption section 308 of the record regenerator cipher-processing section 302, or the data received from means of communications 600 through the communications department 305 is performed. By the format type 3, since the contents key Kcon does not exist in data, decryption processing of the contents key Kcon is not performed.

[0343] At the following step S107, decode of block information is performed using the block information key Kbit decoded at step S161, and the control section 306 of the record regenerator cipher-processing section 302 generates the check value B

(ICVb') from the block information key Kbit and block information (BIT) in step S162 further. The check value B uses as a key the check value B generation key Kicvb saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates the exclusive-OR value which consists of a block information key Kbit and block information (BIT) by DES. Next, in step S109, ICVb in the check value B and a header (Header) is compared, and when in agreement, it progresses to step S151.

[0344] By the format type 3, the check values B and ICVb function as a check value for verifying the alteration of the block information key Kbit and block information. When in agreement with check value:ICVb by which the generated check value B was stored in the header (Header), it is judged that there is no alteration of the block information key Kbit and block information.

[0345] Since steps S151-S112 are the same as that of processing of the format type 2, explanation is omitted.

[0346] At step S163, it decodes with the delivery key Kdis which generated the block key Kblc contained in the contents data read at step S151 at step S105.

[0347] Next, the record regenerator cipher-processing section 302 of the record regenerator 300 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 encipher the block information key Kbit decrypted at step S161, and the block key Kblc decoded at step S163 at step S164 with the session key Kses shared on the occasion of mutual recognition. The control section 301 of the record regenerator 300 reads the block information key Kbit and the block key Kblc which were enciphered with the session key Kses from the record regenerator cipher-processing section 302 of the record regenerator 300, and transmits these data to a storage device 400 through the record device controller 303 of the record regenerator 300.

[0348] Next, the storage device 400 which received the block information key Kbit transmitted from the record regenerator 300, and the block key Kblc in step S165 The received data in the code / decryption section 406 of the storage device cipher-processing section 401 It is made to decrypt with the session key Kses shared on the occasion of mutual recognition. It is made to re-encipher with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401. The control section 301 of the record regenerator 300 The block information key Kbit and the block key Kblc which were re-enciphered with the preservation key Kstr are read from a storage device 400 through the record device controller 303 of the record regenerator 300. That is, the

block information key Kbit enciphered with the delivery key Kdis and the block key Kblc are transposed to the block information key Kbit and the block key Kblc which were re-enciphered with the preservation key Kstr at the beginning.

[0349] Since the following steps S119-S122 are the same as that of the above-mentioned format types 0, 1, and 2, explanation is omitted.

[0350] The above is the mode of download processing of the contents data in the format type 3. It is the format which was suitable for data processing as which quick processing is attained and real-time operations, such as music data, are required since download processing of the format type 3 did not perform decode of block data, and inquiry processing of a contents check value like the format type 2 as mentioned above. Moreover, since the range which protects encryption contents with the block key Kblc is localized, as compared with the format type 2, security serves as altitude more.

[0351] Next, the regeneration from the storage device 400 in the record regenerator 300 in zero to format type 3 each is explained using the flow of drawing 42 -45.

[0352] First, regeneration of the contents in the format type 0 is explained using drawing 42 .

[0353] Step S201 is an authentication processing step between a record regenerator and a storage device, and is performed according to the authentication processing flow of drawing 20 explained previously.

[0354] If authentication processing of step S201 is completed and an authentication flag is set, in step S202, the record regenerator 300 will read the header of the data according to a predetermined format from a storage device 400, and will transmit it to the record regenerator cipher-processing section 302 of the record regenerator 300.

[0355] Next, the control section 306 of the cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the check value A in step S203. As shown in drawing 23 explained previously, the check value A uses as a key the check value A generation key Kicva saved at the internal memory 307 of the record regenerator cipher-processing section 302, and identification information (Content ID) and a handling plan (Usage Policy) are calculated as a message. Next, in step S204, check value:ICVa stored in the calculated check value A and the header (Header) is compared, and when in agreement, it progresses to step S205.

[0356] The check values A and ICVa are check values for verifying the alteration of identification information and a handling plan. When in agreement with check value:ICVa by which the calculated check value A was stored in the header (Header),

it is judged that there is no alteration of the identification information stored in the storage device 400 and a handling plan.

[0357] Next, in step S205, the control section 301 of the record regenerator 300 takes out the block information key Kbit and the contents key Kcon which were enciphered with the preservation key Kstr of a storage device proper from the read header, and transmits to a storage device 400 through the record device controller 303 of the record regenerator 300.

[0358] The code / decryption section 406 of the storage device cipher-processing section 401 are made to carry out decryption processing with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401, and the storage device 400 which received the block information key Kbit transmitted from the record regenerator 300 and the contents key Kcon makes it encipher the received data again with the session key Kses shared on the occasion of mutual recognition. This processing is as the column of the key message exchange after (9) mutual recognition mentioned above having described in detail.

[0359] At step S206, the control section 301 of the record regenerator 300 receives the block information key Kbit and the contents key Kcon which were re-enciphered with the session key Kses from the storage device 400 through the record device controller 303 of the record regenerator 300.

[0360] In step S207 next, the control section 301 of the record regenerator 300 The block information key Kbit and the contents key Kcon which were re-enciphered with the received session key Kses are transmitted to the record regenerator cipher-processing section 302 of the record regenerator 300. The record regenerator cipher-processing section 302 of the record regenerator 300 which received the block information key Kbit re-enciphered with the session key Kses and the contents key Kcon The code / decryption section 308 of the record regenerator cipher-processing section 302 are made to decrypt the block information key Kbit enciphered with the session key Kses, and the contents key Kcon with the session key Kses shared on the occasion of mutual recognition.

[0361] Furthermore, in step S208, the block information read at step S202 with the decrypted block information key Kbit is decrypted. In addition, the decrypted block information key Kbit, the contents key Kcon, and the block information BIT are transposed to the block information key Kbit contained in the header read at step S202, the contents key Kcon, and the block information BIT, and the record regenerator cipher-processing section 302 of the record regenerator 300 holds them.

Moreover, the control section 301 of the record regenerator 300 reads the decrypted block information BIT from the record regenerator cipher-processing section 302 of the record regenerator 300.

[0362] Furthermore, in step S209, the control section 306 of the record regenerator cipher-processing section 302 generates the check value B (ICVb') from the block information key Kbit, the contents key Kcon, and block information (BIT). As shown in drawing 24 , the check value B uses as a key the check value B generation key Kicvb saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates the exclusive-OR value which consists of the block information key Kbit, a contents key Kcon, and block information (BIT) by DES. Next, in step S210, ICVb in the check value B and a header (Header) is compared, and when in agreement, it progresses to step S211.

[0363] It is judged that the alteration of the block information key Kbit in the data saved at the storage device 400, the contents key Kcon, and block information does not have check values B and ICVb when in agreement with check value:ICVb by which the check value B which is a check value for verifying the alteration of the block information key Kbit, the contents key Kcon, and block information, and was generated was stored in the header (Header).

[0364] The control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate a middle check value in step S211. As shown in drawing 25 , a middle check value uses as a key the total check value generation key Kicvt saved at the internal memory 307 of the record regenerator cipher-processing section 302, and calculates it according to the ICV count approach explained by drawing 7 etc. by making the check value A in verified Header, the check value B, and all the contents check values in block information into a message. In addition, the generated middle check value is held in the record regenerator cipher-processing section 302 of the record regenerator 300 if needed.

[0365] In step S212 next, the control section 301 of the record regenerator 300 Use limit information is taken out from the handling plan (Usage Policy) contained in the header unit of the data read from the external memory 402 of a storage device 400. the contents of a playback schedule -- the record regenerator 300 concerned -- it can use (use limit information is 1) -- same another record regenerator 300 -- it can use (use limit information is 0) -- it judges. When the use limit which 1, i.e., playback contents, can use [use limit information] only with the record regenerator 300 concerned as a result of a judgment is set up, it progresses to step S213, and when it

is a setup which use limit information can use by 0 [300], i.e., same another record regenerator, it progresses to step S215. In addition, the cipher-processing section 302 may perform processing of step S212.

[0366] The control section 301 of the record regenerator 300 makes the record regenerator cipher-processing section 302 of the record regenerator 300 calculate check value ICVdev' of a record regenerator proper at step S213. Check value ICVdev' of a record regenerator proper uses as a key the record regenerator signature key Kdev saved at the internal memory 307 of the record regenerator cipher-processing section 302, as shown in drawing 25 , and it enciphers and generates the middle check value held at step S211 by DES.

[0367] Next, in step S214, ICVdev in the header read at check value ICVdev' of the record regenerator proper calculated at step S213 and step S202 is compared, and when in agreement, it progresses to step S217.

[0368] On the other hand, the control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate the total check value ICVt at step S215. As shown in drawing 25 , total check value ICVt' uses as a key the system signature key Ksys saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates a middle check value by DES. Next, in step S216, ICVt in generated total check value ICVt' and a header (Header) is compared, and when in agreement, it progresses to step S217.

[0369] The total check value ICVt and the check value ICVdev of a record regenerator proper are check values for verifying the alteration of all the check values of ICVa, ICVb, and each contents block. Therefore, when in agreement with check value:ICVt or ICVdev by which the check value generated by above-mentioned processing was stored in the header (Header), it is judged that there is no alteration of all the check values of ICVa and ICVb which were stored in the storage device 400, and each contents block.

[0370] Next, in step S217, the control section 301 of the record regenerator 300 reads block data from a storage device 400. Furthermore, it judges whether it is enciphered in step S218, and when enciphered, block data is decoded in the cipher-processing section 302 of the record regenerator 300. When not enciphered, step S219 is skipped and it progresses to step S220.

[0371] Next, in step S220, the control section 301 of the record regenerator 300 investigates whether there is any paddle with which the contents block is a candidate for verification based on the contents block information within block information (BIT).

When the contents block is a candidate for verification, the contents check value is stored in the block information in a header. When the contents block has become a candidate for verification, contents check value ICVi' of the corresponding contents block is made to calculate in step S221. When the contents block is not a candidate for verification, steps S221 and S222 are skipped and it progresses to step S223.

[0372] When the block is enciphered as drawing 36 explained previously, contents check value ICVi' is the contents key Kcon, decrypts the inputted contents block in the CBC mode of DES, and enciphers and generates the contents mean value which carried out the exclusive OR of the whole of the result per 8 bytes, and generated it with the contents check value generation key Kicvc in which it was stored by the internal memory 307 of the record regenerator 300. Moreover, when the block is not enciphered, it is generated as a value which inputted the whole data (plaintext) into the alteration check value generating function (let DES-CBC-MAC and the contents check value generation key Kicvc be keys) shown in drawing 36 per 8 bytes, and obtained it.

[0373] In step S222, the control section 306 of the record regenerator cipher-processing section 302 compares the contents check value ICVi stored in the header unit which received from the storage device 400 at step S202 with generated contents check value ICVi', and passes the result to the control section 301 of the record regenerator 300. The control section 301 of the record regenerator 300 which received this stores the contents plaintext data for activation (playback) on the record regenerator system RAM in step S223, when having succeeded in verification. The same verification processing and RAM storing processing are repeated until the control section 301 of the record regenerator 300 takes out the following contents block for verification further, it makes the record regenerator cipher-processing section 302 of the record regenerator 300 verify it and it verifies all contents blocks (step S224).

[0374] In addition, in step S204, step S210, step S214, step S216, or step S222, when coincidence of a check value is not obtained, regeneration is ended as an error.

[0375] If judged with whole block read-out in step S224, it will progress to step S225 and activation of contents (a program, data) and playback will be started.

[0376] The above is the mode of regeneration of the contents data in the format type 0.

[0377] Next, regeneration of the contents data in the format type 1 is explained using drawing 43 . It explains focusing on a different point from regeneration above-mentioned format type 0.

[0378] Since the processing to step S201 - step S217 is the same as that of regeneration above-mentioned format type 0, explanation is omitted.

[0379] By the format type 1, in step S231, decode of encryption parts is performed and Parts ICV are generated. Furthermore, block ICVi' is generated in step S232. As explained previously, when at least one of the parts within a block is object data of the check value ICVi, in the format type 1, the contents check value ICVi is defined about the block. Check value P-ICVij of the parts j in Block i is generated as a value which enciphered the value which carried out exclusive OR of the whole plaintext (decode sentence) per 8 bytes with the contents check value generation key Kicvc, when Parts j are enciphered. Moreover, when Parts j are not enciphered, it is generated as a value which inputted the whole data (plaintext) into the alteration check value generating function (let DES-CBC-MAC and the contents check value generation key Kicvc be keys) shown in drawing 36 per 8 bytes, and obtained it.

[0380] Furthermore, when one parts which are [ICV flag =subject of ICV] which shows that it is a candidate for a check exist in one block i Check value P-ICVij generated by above-mentioned technique is made into the check value ICVi of a block as it is. Moreover, when two or more parts which are [ICV flag =subject of ICV] which shows that it is a candidate for a check exist in one block i Two or more parts check value P-ICVi, the alteration check value generating function which shows the whole data (plaintext) to drawing 36 per 8 bytes for the data which connected j with the parts numerical order (DES-CBC-MAC) It is generated as a value which inputted the contents check value generation key Kicvc for considering as a key, and obtained it. This is as drawing 37 having explained previously.

[0381] In the format type 1, comparison processing of the contents check value generated in the above-mentioned procedure will be performed at step S222. Since the following processings not more than step S223 are the same as that of the format type 0, explanation is omitted.

[0382] Next, regeneration of the contents data in the format type 2 is explained using drawing 44 . It explains focusing on a different point from regeneration above-mentioned format type 0 and 1.

[0383] Since steps S201-S210 are the same as that of regeneration above-mentioned format type 0 and 1, explanation is omitted.

[0384] In the format type 2, processing of steps S211-S216 performed in the format types 0 and 1 is not performed. Moreover, in the format type 2, since it does not have a contents check value, verification of the contents check value of step S222 performed in the format types 0 and 1 is not performed, either.

[0385] In format type 2 data regeneration, it progresses to step S217 after the verification step of the check value B of step S210, and block data is read by control of the control section 301 of the record regenerator 300. Furthermore, in step S241, decode processing of the block key Kblc contained in the block data based on the cipher-processing section 306 of the record regenerator 300 is performed. As drawing 34 shows, it is enciphered with the contents key Kcon, and the block key Kblc stored in the storage device 400 decodes the block key Kblc using the contents key Kcon decoded in previous step S207.

[0386] Next, in step S242, decode processing of block data is performed using the block key Kblc decoded at step S241. Furthermore, activation of contents (a program, data) and regeneration are performed in step S243. Processing of step S217 - step S243 is repeatedly performed about a whole block. Regeneration will be ended if judged with whole block read-out in step S244.

[0387] Thus, processing of the format type 2 is the configuration that are omitting check value verification processing of the total check value etc., and it is suitable for activation of high-speed decode processing, and is the format suitable for data processing as which real-time operations, such as music data, are required.

[0388] Next, regeneration of the contents data in the format type 3 is explained using drawing 45 . It explains focusing on a different point from regeneration above-mentioned format type 0, 1, and 2.

[0389] Although the format type 3 has many parts which are fundamentally common in the processing in the format type 2, as the format type 3 was explained in drawing 35 , it does not have a contents key, and differs from the format type 2 in that the block key Kblc is enciphered and stored with the preservation key Kstr in a storage device.

[0390] In steps S201-S210, processing of step S251, step S252, step S253, and step S254 is constituted as processing which does not contain a contents key unlike the correspondence processing in the above-mentioned format types 0, 1, and 2.

[0391] In step S251, the control section 301 of the record regenerator 300 takes out the block information key Kbit enciphered with the preservation key Kstr of a storage device proper from the read header, and transmits to a storage device 400 through the record device controller 303 of the record regenerator 300.

[0392] The code / decryption section 406 of the storage device cipher-processing section 401 are made to carry out decryption processing with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401, and the storage device 400 which received the block information key Kbit transmitted from the record regenerator 300 makes it

re-encipher the received data with the session key Kses shared on the occasion of mutual recognition. This processing is as the column of the key message exchange after (9) mutual recognition mentioned above having described in detail.

[0393] At step S252, the control section 301 of the record regenerator 300 receives the block information key Kbit re-enciphered with the session key Kses from the storage device 400 through the record device controller 303 of the record regenerator 300.

[0394] In step S253 next, the control section 301 of the record regenerator 300 The block information key Kbit re-enciphered with the received session key Kses is transmitted to the record regenerator cipher-processing section 302 of the record regenerator 300. The record regenerator cipher-processing section 302 of the record regenerator 300 which received the block information key Kbit re-enciphered with the session key Kses The code / decryption section 308 of the record regenerator cipher-processing section 302 are made to decrypt the block information key Kbit enciphered with the session key Kses with the session key Kses shared on the occasion of mutual recognition.

[0395] Furthermore, in step S208, the block information read at step S202 with the decrypted block information key Kbit is decrypted. In addition, the decrypted block information key Kbit and the block information BIT are transposed to the block information key Kbit contained in the header read at step S202, and the block information BIT, and the record regenerator cipher-processing section 302 of the record regenerator 300 holds them. Moreover, the control section 301 of the record regenerator 300 reads the decrypted block information BIT from the record regenerator cipher-processing section 302 of the record regenerator 300.

[0396] Furthermore, in step S254, the control section 306 of the record regenerator cipher-processing section 302 generates the check value B (ICVb') from the block information key Kbit and block information (BIT). As shown in drawing 24 , the check value B uses as a key the check value B generation key Kicvb saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates the exclusive-OR value which consists of a block information key Kbit and block information (BIT) by DES. Next, in step S210, ICVb in the check value B and a header (Header) is compared, and when in agreement, it progresses to step S211.

[0397] By the format type 3, further, since a block key is enciphered with a preservation key at the time of storing with a storage device, decode processing with the session key in the record regenerator 300 is needed for decode processing with the preservation key in a storage device 400 and encryption processing with a session

key, and a pan. These the processings of a series of are the processing steps shown at step S255 and step S256.

[0398] At step S255, the control section 301 of the record regenerator 300 takes out the block key Kblc enciphered with the preservation key Kstr of a storage device proper from the block read at step S217, and transmits to a storage device 400 through the record device controller 303 of the record regenerator 300.

[0399] The code / decryption section 406 of the storage device cipher-processing section 401 are made to carry out decryption processing with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401, and the storage device 400 which received the block key Kblc transmitted from the record regenerator 300 makes it re-encipher the received data with the session key Kses shared on the occasion of mutual recognition. This processing is as the column of "the key message exchange after (9) mutual recognition" mentioned above having described in detail.

[0400] At step S256, the control section 301 of the record regenerator 300 receives the block key Kblc re-enciphered with the session key Kses from the storage device 400 through the record device controller 303 of the record regenerator 300.

[0401] Next, in step S257, decode processing using the session key Kses of the block key Kblc by the cipher-processing section 306 of the record regenerator 300 is performed.

[0402] Next, in step S242, decode processing of block data is performed using the block key Kblc decoded at step S257. Furthermore, activation of contents (a program, data) and regeneration are performed in step S243. Processing of step S217 – step S243 is repeatedly performed about a whole block. Regeneration will be ended if judged with whole block read-out in step S244.

[0403] The above processing is regeneration of the contents in the format type 3. Although it is similar with the format type 2 in that verification processing of the total check value was omitted, as compared with the format type 2, it has high processing composition of security level further at the point including the key message exchange of a block key.

[0404] (11) It has explained that verification processing about various kinds of check values ICV is performed in phases, such as download of contents, or regeneration, in the check value (ICV) generation processing mode above-mentioned example in a content provider. Here, the mode of each [these] check value (ICV) generation processing and verification processing is explained.

[0405] First, about each check value explained in the example, when it collects briefly,

there are the following in the check value ICV used in the data processor of this invention.

[0406] Check values A and ICVa: The check value for verifying the alteration of the identification information in contents data, and a handling plan.

Check values B and ICVb: The check value for verifying the alteration of the block information key Kbit, the contents key Kcon, and block information.

Contents check value ICVi: The check value for verifying the alteration of each contents block of contents.

The total check value ICVt: It is a check value for verifying the alteration of the check value ICVa, the check value ICVb, and all the check values of each contents block.

Regenerator proper check value ICVdev: When the localization flag is set to 1 (i.e., when it is shown that contents are available to a record regenerator proper), it is the check value transposed to the total check value ICVt, and is generated as a check value over above-mentioned check value A:ICVa, check value B:ICVb, and the whole check value ICVi contained in each block further set as the check object of contents. It may become the contents itself instead of the check value of each contents block to be contained in the object which ICVt and ICVdev check depending on a format.

[0407] Each above check value is used in the data processor of this invention. In each above-mentioned check value, an ICV value is generated by drawing 32 -35 and the content provider who offers contents data as shown in drawing 6 , or the contents manager based on each data for verification, the check value A, the check value B, the total check value, and a contents check value are stored in data with contents, and the user of the record regenerator 300 is provided with them. In case [in which these contents are downloaded to a storage device] it reproduces in the case, ICV for verification is generated based on each data for verification, the user, i.e., the contents user, of a record regenerator, and he performs the comparison with ICV [finishing / storing]. Moreover, when it is shown that contents are available to a record regenerator proper, the regenerator proper check value ICVdev is transposed to the total check value ICVt, and is stored in a storage device.

[0408] Generation processing of a check value has mainly explained the generation processing configuration by DES-CBC in the above-mentioned example. However, there are not only an above-mentioned approach but various generation processing modes and still more various verification processing modes among the generation processing modes of ICV. Various kinds of ICV generation and verification processing configurations which are especially explained below in the relation between a contents provider or a manager, and a contents user are possible.

[0409] Drawing which explains the generation processing in the generation person of the check value ICV and the verification processing by the verification person to drawing 46 – drawing 48 is shown.

[0410] Drawing 46 is the configuration of the ICV generation person who is for example, a contents provider or a manager performing generation processing of ICV by DES-CBC explained in the above-mentioned example, and providing a record regenerator user, i.e., a verification person, with generated ICV with contents. In this case, the key which is needed in case a record regenerator user, i.e., a verification person, is verification processing is each check value generation key stored in the internal memory 307 shown in drawing 18 . The verification person (record regenerator user) who is a contents user uses the check value generation key stored in the internal memory 307, generates a check value with the application of DES-CBC to the data for verification, and performs comparison processing with a storing check value. In this case, each check value generation key is constituted as a key which the generation person and verification person of ICV share secretly.

[0411] A contents user, i.e., a verification person, is provided with ICV which the generation person of ICV whose drawing 47 is a contents provider or a manager generated ICV by the digital signature of a public-key-encryption system, and generated with contents. A contents user, i.e., a verification person, is the configuration of saving an ICV generation person's public key and performing verification processing of ICV using this public key. In this case, it is not necessary to make secret an ICV generation person's public key which a contents user (record regenerator user), i.e., a verification person, has, and management becomes easy. It is the mode which was suitable when carried out on security management level with generation of ICV and management high when generation of ICV and management are performed in one entity.

[0412] The generation person of ICV which is a contents provider or a manager generates ICV by the digital signature of a public-key-encryption system, and provides a contents user, i.e., a verification person, with generated ICV with contents, and further, a verification person stores in a public key certificate (for example, refer to drawing 14) the public key used for verification, and provides a record regenerator user, i.e., a verification person, with drawing 48 with contents data. When two or more generation persons of ICV exist, each generation person has a key management center create the data (public key certificate) proving the justification of a public key.

[0413] If the contents user who is a verification person of ICV has the public key of a key management center, a verification person performs verification of a public key

certificate with the public key of a key management center and justification is checked, the public key of the generation person of ICV stored in the public key certificate will be taken out. Furthermore, verification of ICV is performed using the taken-out public key of the generation person of ICV.

[0414] This approach is an effective mode when the executive system of management by the center in which the generation person of ICV performs those with two or more and those managements is established.

[0415] (12) Explain the generation configuration of the various keys for cipher processing based on a master key which is one of the cipher-processing key generation configuration based on a master key, next the characteristic configurations in the data processing system of this invention.

[0416] As previously explained using drawing 18 , various master keys are stored in the internal memory of the record regenerator 300 in the data processor of this invention, and using each of these master keys, the authentication key Kake is generated (several 3 reference), or it has composition which generates the delivery key Kdis (several 4 reference).

[0417] In case cryptocommunication, mutual recognition, MAC generation, verification, etc. were conventionally performed between the record regenerators 300 and archive media 400 between the entities of 1 to 1 (i.e., between a content provider and a contents user), or in the data processor of above-mentioned this invention, confidential information common to each entity, for example, key information, was made to hold. Moreover, it sets in relations, such as an archive medium of a large number to the relation of one-pair **, for example, many contents users to one content provider, and one record regenerator. [whether it considers as the configuration which carries out storing maintenance of the confidential information made to share in all entities, i.e., many contents users, or many archive media, for example, the key information, and] or one content provider managed much each contents user's confidential information (ex. key) according to the individual, and used this properly according to each contents user.

[0418] However, in the configuration which owns the confidential information (ex. key) which all share, when there is use relation of the above one-pair **, when one secret leakage occurs, there is a fault that effect attains to all the persons using the same confidential information (ex. key). moreover, the list which one manager, for example, a content provider, managed much each contents user's confidential information (ex. key) according to the individual, identified all the users when it was the configuration which uses this properly according to each contents user, and matched the

confidential information (ex. key) of a proper with the discernment data is needed, and there is a fault that the burden of the maintenance control of the list accompanying a user's increase increases.

[0419] It set to the data processor of this invention, and the conventional trouble in sharing of the confidential information between such entities was solved by possession of a master key, and the configuration which generates various kinds of individual keys from a master key. Hereafter, this configuration is explained.

[0420] In the data processor of this invention, when the key according to different individual in various kinds of cipher processing between the media which stored a storage device and contents, or a record regenerator, authentication processing, etc. is needed, a device and media generate the key according to the individual using the individual key generation method decided beforehand within the individual information on the identifier data (ID) which it has in a proper, and the record regenerator 300. If leakage of a master key is prevented even when the key according to individual which should have been generated by this configuration is specified, it will become possible to prevent system-wide damage. Moreover, it matches by the configuration which generates a key with a master key, and management of a list also becomes unnecessary.

[0421] The concrete example of a configuration is explained using drawing. First, drawing explaining the configuration which generates various kinds of keys using various kinds of master keys which the record regenerator 300 has to drawing 49 is shown. From the media 500 of drawing 49 , and means of communications 600, contents are inputted like the already explained example. Contents are enciphered with the contents key Kcon, and the contents key Kcon is enciphered with the delivery key Kdis.

[0422] For example, when the record regenerator 300 tends to take out contents from media 500 and means of communications 600 and tends to download to a storage device 400, as explained in previous drawing 22 and drawing 39 -41, it is necessary for the record regenerator 300 to acquire the delivery key Kdis which has enciphered the contents key. This Kdis is directly acquired from media 500 and means of communications 600, or although it is also possible for the record regenerator 300 to acquire beforehand and to store in the memory in the record regenerator 300, the distribution configuration to many users of such a key has the possibility of the leakage which affects the whole system as explained also in advance.

[0423] In the data processing system of this invention, it is carrying out to the processing based on the master key MKdis for delivery keys and content ID in which

this delivery key Kdis was stored by the memory of the record regenerator 300 as shown in the lower part of drawing 49 , i.e., the configuration which generates the delivery key Kdis with the application of Kdis=DES (MKdis, content ID). Even if it is the case where many content providers exist in the contents distribution configuration between the record regenerators 300 which are the content provider who supplies contents, and its contents user from media 500 and means of communications 600 according to this configuration, it is not necessary to circulate each delivery key Kdis through media, communication media, etc., and to store in each record regenerator 300, and it becomes possible to maintain security at altitude.

[0424] Next, generation of the authentication key Kake is explained. When setting the contents stored in the download processing to an archive medium 400 or drawing 28 , and the archive medium 400 explained by drawing 42 -45 to the record regenerator 300 and performing and reproducing from drawing 22 and the record regenerator 300 of drawing 39 -41 which were explained previously, the mutual recognition processing between the record regenerator 300 and an archive medium 400 (refer to drawing 20) is needed.

[0425] As drawing 20 explained, in this authentication processing, as for the record regenerator 300, the authentication key Kake is needed. An authentication key is directly acquired from an archive medium 400, or although it is also possible for the record regenerator 300 to acquire the record regenerator 300 beforehand, and to store in the memory in the record regenerator 300, the distribution configuration to many users of such a key as well as the configuration of an above-mentioned delivery key has the possibility of the leakage which affects the whole system.

[0426] In the data processing system of this invention, it is carrying out to the processing based on the master key MKake for authentication keys and storage device discernment ID:IDmem in which this authentication key Kake was stored by the memory of the record regenerator 300 as shown in the lower part of drawing 49 , i.e., the configuration which asks for the authentication key Kake by Kake=DES (MKake, IDmem).

[0427] Furthermore, drawing 22 , download processing to the archive medium 400 from the record regenerator 300 of drawing 39 -41, The contents stored in drawing 28 and the archive medium 400 explained by drawing 42 -45 are set to the record regenerator 300. Or activation, When reproducing, it can consider as the configuration same also about the record regenerator signature key Kdev which is needed for generation processing of the record regenerator proper check value ICVdev in the case of being contents available to a record regenerator proper as an

above-mentioned delivery key and an authentication key. Although the record regenerator signature key Kdev was considered as the configuration stored in an internal memory in the above-mentioned example Store the master key MKdev for record regenerator signature keys in memory, and the record regenerator signature key Kdev is not stored in an internal memory. As shown in the lower part of drawing 49 if needed, it is based on record regenerator identifier:IDdev and the master key MKdev for record regenerator signature keys. By considering as the configuration which asks for the record regenerator signature key Kdev by Kdev=DES (MKdev, IDdev), the advantage of it becoming unnecessary to give the record regenerator signature key Kdev according to a device individual is mentioned.

[0428] Thus, it sets to the data processor of this invention. Since information on a key required for the procedure about the code information processing between two entities like [between a provider, a record regenerator or a record regenerator, and a storage device] etc. was considered as the configuration which generates on a target serially from a master key and each ID Even when key information is revealed from each entity, the range of damage with the key according to individual becomes unnecessary [management of the key list for every entity according to individual which it was limited more and mentioned above].

[0429] A flow is shown and two or more examples of processing related with this configuration are explained. Drawing 50 is an example of decode processing, encryption processing of the contents using the master key in contents manufacture or a manager etc., and a user device, for example, the encryption data using the master key in the record regenerator 300 in an above-mentioned example.

[0430] Step S501 in contents manufacture or a manager is a step which gives the identifier (content ID) to contents. Step S502 is a step which generates the key which enciphers contents etc. based on the master key and content ID which contents manufacture or a manager has. This generates the delivery key Kdis by the process which generates for example, the delivery key Kdis, then above-mentioned Kdis=DES (MKdis, content ID). Next, step S503 is a step which enciphers a part or all of contents with a key (for example, the delivery key Kdis). A contents manufacturer distributes the contents which performed encryption processing through such a step through media, such as DVD, means of communications, etc.

[0431] On the other hand, in the user device side of for example, record regenerator 300 grade, content ID is read in step S504 out of the contents data received through media, means of communications, etc. Next, in step S505, the read content ID and the key applied to decode of encryption contents based on the master key to own are

generated. This generation processing serves as delivery key Kdis=DES (MKdis, content ID), when the delivery key Kdis is obtained. At step S506, contents are decoded using this key, and use, i.e., playback, or the program of decode contents is performed at step S507.

[0432] In this example, as shown in the drawing 50 lower berth, contents manufacture or a manager, and the both sides of a user device have a master key (for example, the master key MKdis for delivery key generation), and generate serially a delivery key required for encryption of contents, and decode based on each master key to own and each ID (content ID) on a target.

[0433] In this system, when a delivery key should be revealed to a third person, decode of those contents becomes possible in a third person, but since decode of other contents from which content ID differs can be prevented, it is effective in the ability to make into the minimum effect leakage of one contents key affects the whole system. Moreover, in a user device side, i.e., a record regenerator, it is effective in not holding the matching list of keys for every contents.

[0434] Next, using drawing 51 , contents manufacture or a manager owns two or more master keys, and explains the example which performs processing according to the candidate for distribution of contents.

[0435] Step S511 in contents manufacture or a manager is a step which gives the identifier (content ID) to contents. Step S512 is a step which chooses one master key from two or more master keys (for example, two or more master keys MKdis for delivery key generation) which contents manufacture or a manager has. Although further explained using drawing 52 , this selection processing sets up the master key which matches for every version of every country of the user of contents, every model, and a model, and is applied beforehand, and is performed according to that setup.

[0436] Next, at step S513, the key for encryption is generated based on the master key chosen at step S512, and the content ID determined at step S511. The process which generates for example, the delivery key Kdisi, then Kdisi=DES (MKdisi, content ID) generate this. Next, step S514 is a step which enciphers a part or all of contents with a key (for example, the delivery key Kdisi). A contents manufacturer distributes content ID, the used master key identification information, and the contents which performed encryption processing for encryption contents as one distribution unit through media, such as DVD, means of communications, etc. in step S515.

[0437] On the other hand, in the user device side of for example, record regenerator 300 grade, it judges whether self owns the master key corresponding to the master

key identification information in the contents data distributed through media, such as DVD, means of communications, etc. in step S516. When it does not have a master key corresponding to the master key identification information in contents data, the distribution contents cannot be used in the user device, and processing is ended.

[0438] When self owns the master key corresponding to the master key identification information in the distributed contents data, in step S517, content ID is read out of the contents data received through media, means of communications, etc. Next, in step S518, the read content ID and the key applied to decode of encryption contents based on the master key to own are generated. This generation processing serves as delivery key Kdisi=DES (MKdisi, content ID), when the delivery key Kdisi is obtained. At step S519, contents are decoded using this key, and use, i.e., playback, or the program of decode contents is performed at step S520.

[0439] In this example, as shown in the drawing 51 lower berth, contents manufacture or a manager has the master key set which consists of two or more master keys MKdis1-n, for example, two or more master keys for delivery key generation. On the other hand, only when it has one master key KKdisi, for example, one master key for delivery key generation, in a user device and contents manufacture or a manager is doing encryption processing using MKdisi, a user device can decode the contents and can be used.

[0440] The example which applied a different master key for every country as an example of a mode shown in the flow of this drawing 51 is shown in drawing 52 . A content provider has the master keys MK1-n, and uses MK1 for the key generation which performs encryption processing of the contents distributed to the user device for Japan. For example, the encryption key K1 is generated from content ID and MK1, and contents are enciphered by K1. Moreover, MK2 was used for the key generation which performs encryption processing of the contents distributed to the user device for USs, and it has set up MK3 so that it may use for the key generation which performs encryption processing of the contents distributed to the user device turned EU (Europe).

[0441] On the other hand, the master key MK1 is stored in the internal memory at record regenerators, such as a user device for Japan, PC specifically sold in Japan, or a game device, the master key MK2 is stored in the internal memory at the user device for USs, and the master key MK3 is stored in the user device for EU at the internal memory.

[0442] In such a configuration, a content provider performs encryption processing of the contents which distribute contents to a user device from the master keys MK1-n

according to an available user device, using a master key alternatively. For example, in order to make contents available [the user device for Japan], contents are enciphered with the key K1 generated using the master key MK1. Although these encryption contents can generate the decode possibility of, i.e., a decode key, using the master key MK1 stored in the user device for Japan, since they cannot obtain a key K1 from the master keys MK2 and MK3 stored in other USs or the user device for EU, the decode of encryption contents of them becomes impossible.

[0443] Thus, when a content provider uses two or more master keys alternatively, a use limit of various contents can be set up. Although drawing 52 showed the example which distinguishes a master key according to the country of a user device, various use gestalten, such as switching a master key according to a version, corresponding to the model of user device, are possible as mentioned above.

[0444] Next, the example of processing which combined the identifier, i.e., Media ID and a master key, of a media proper with drawing 53 is shown. It is here. Media are the media which stored contents, such as DVD and CD. It is good also as a proper for each Media of every, and Media ID are good also as a proper for every title of contents, such as a movie, and good also as a proper for every manufacture lot of media, for example. Thus, various approaches as an approach to assign Media ID can be used.

[0445] Step S521 in media manufacture or a manager is a step which determines the identifier (media ID) to media. Step S522 is a step which generates the master key which media manufacture or a manager has, and the key which enciphers the storing contents in media etc. based on Media ID. This generates the delivery key Kdis by the process which generates for example, the delivery key Kdis, then above-mentioned Kdis=DES (MKdis, media ID). Next, step S523 is a step which enciphers a part or all of media storing contents with a key (for example, the delivery key Kdis). A media manufacturer supplies the contents storing media which performed encryption processing through such a step.

[0446] On the other hand, in the user device side of for example, record regenerator 300 grade, Media ID are read from the supplied media in step S524. Next, in step S525, the read media ID and the key applied to decode of encryption contents based on the master key to own are generated. This generation processing serves as delivery key Kdis=DES (MKdis, media ID), when the delivery key Kdis is obtained. At step S526, contents are decoded using this key, and use, i.e., playback, or the program of decode contents is performed at step S527.

[0447] In this example, as shown in the drawing 53 lower berth, media manufacture or a manager, and the both sides of a user device have a master key (for example, the

master key MKdis for delivery key generation), and generate serially a delivery key required for encryption of contents, and decode based on each master key to own and each ID (media ID) on a target.

[0448] In this system, when a media key should be revealed to a third person, decode of the contents in those media becomes possible in a third person, but since decode of the contents stored in other media from which Media ID differ can be prevented, it is effective in the ability to make into the minimum effect leakage of one media key affects the whole system. Moreover, in a user device side, i.e., a record regenerator, it is effective in not holding the matching list of keys for every media. Moreover, since the contents size enciphered with one media key is restricted to a capacity storable in the media, there can be little possibility of reaching amount of information required for a cipher attack, and can reduce the possibility of decryption.

[0449] Next, the example of processing which combined the identifier, i.e., the record regenerator ID and a master key, of a record regenerator proper with drawing 54 is shown.

[0450] Step S531 in a record regenerator user is a step which generates the master key stored in the internal memory of a record regenerator, and the key which enciphers contents etc. based on the record regenerator ID. This generates the contents key Kcon by the process which generates for example, the contents key Kcon, then Kcon=DES (MKcon, the record regenerator ID). Next, step S532 is a step which enciphers a part or all of contents that is stored with a key (for example, the delivery key Kcon). Step S533 stores encryption contents in storage devices, such as a hard disk.

[0451] On the other hand, in the system administrator side who manages a record regenerator, if restoration of storing data is requested from the record regenerator user who stored contents, in step S534, the record regenerator ID will be read from a record regenerator. Next, in step S535, the key applied to decode of encryption contents based on the read record regenerator ID and the master key to own is generated. This generation processing serves as contents key Kcon=DES (MKcon, the record regenerator ID), when the contents key Kcon is obtained. At step S536, contents are decoded using this key.

[0452] In this example, as shown in the drawing 54 lower berth, the both sides of a record regenerator user and a system administrator have a master key (for example, the master key MKcon for contents key generation), and generate serially a delivery key required for encryption of contents, and decode based on each master key to own and each ID (record regenerator ID) on a target.

[0453] In this system, when a contents key should be revealed to a third person, decode of those contents becomes possible in a third person, but since decode of the contents enciphered for [from which the record regenerator ID differs / other] record regenerators can be prevented, it is effective in the ability to make into the minimum effect leakage of one contents key affects the whole system. Moreover, in user device side both, it is effective in not holding the matching list of keys for every contents a system management side.

[0454] Drawing 55 is a configuration which generates the authentication key used for the mutual recognition processing between storage devices, such as a slave device, for example, a memory card etc., and a host device, for example, a record regenerator, based on a master key. Although considered as the configuration which stored the authentication key in the internal memory of a slave device beforehand in the authentication processing (refer to drawing 20) explained previously, it can consider as the configuration which generates this based on a master key at the time of authentication processing as shown in drawing 55 .

[0455] For example, the slave device which is a storage device generates the master key stored in the internal memory of the slave device which is a storage device in step S541, and the authentication key Kake used for mutual recognition processing based on a slave device ID as initialization processing before authentication processing initiation. Kake=DES (MKake, a slave device ID) generates this. Next, the generated authentication key is stored in memory in step S542.

[0456] On the other hand, in a host device side, such as for example, a record regenerator, a slave device ID is read from the storage device with which it was equipped, i.e., a slave device, through means of communications in step S543. Next, in step S544, the authentication key applied to mutual recognition processing based on the read slave device ID and the master key for authentication key generation to own is generated. This generation processing serves as for example, authentication key Kake=DES (MKake, a slave device ID). At step S545, authentication processing is performed using this authentication key.

[0457] In this example, as shown in the drawing 55 lower berth, the both sides of a slave device and a master device have the master key MKake, i.e., the master key for authentication key generation, and generate serially an authentication key required for mutual recognition processing based on each master key and slave device ID to own on a target.

[0458] In this system, when an authentication key should be revealed to a third person, since it is effective only in that slave device, in relation with other slave devices,

authentication will be materialized and that authentication key is effective in the ability to make into the minimum effect generated by leakage of a key.

[0459] Thus, it set to the data processor of this invention, and information on a key required for the procedure about the code information processing between two entities like [between a content provider, a record regenerator or a record regenerator, and a storage device] etc. was considered as the configuration which generates on a target serially from a master key and each ID. Therefore, even when key information is revealed from each entity, the range of damage with the key according to individual becomes unnecessary [management of the key list for every entity according to individual which it was limited more and mentioned above].

[0460] (13) In the example in which the code reinforcement in cipher processing carried out control ****, cipher processing between the record regenerator 300 and a storage device 400 has mainly explained the example using cipher processing by the single DES configuration previously explained using drawing 7 , in order to make explanation easy to understand. However, it is not limited to the single DES method mentioned above at all, and encryption mode of processing applied in the data processor of this invention can adopt the cipher system according to a required security condition.

[0461] For example, a Triple DES method like the configuration of drawing 8 - drawing 10 explained previously may be applied. For example, in the both sides of the cipher-processing section 302 of the record regenerator 300 shown in drawing 3 , and the cipher-processing section 401 of a storage device 400, the configuration which performs processing corresponding to cipher processing by the Triple DES method which considered as the configuration which can perform a Triple DES method, and was explained by drawing 8 - drawing 10 is possible.

[0462] However, the provider of contents gives priority to processing speed according to contents, may consider the contents key Kcon as the 64-bit key configuration by the single DES method, and may give priority to security, and may consider the contents key Kcon as 128 bits by the Triple DES method, or a 192-bit key configuration. Therefore, it is not desirable to consider the configuration of the cipher-processing section 302 of the record regenerator 300 and the cipher-processing section 401 of a storage device 400 as the configuration which can respond only to the method of a Triple DES method and one of single DES methods. therefore, the cipher-processing section 302 of the record regenerator 300 and the cipher-processing section 401 of a storage device 400 -- Single DES and Triple DES -- the configuration whose correspondence to any method is enabled is desirable.

[0463] However, in order to consider the cipher-processing configuration of the cipher-processing section 302 of the record regenerator 300, and the cipher-processing section 401 of a storage device 400 as the configuration which can perform the both sides of a single DES method and a Triple DES method, each another circuit and logic must be constituted. For example, in order to perform processing corresponding to Triple DES in a storage device 400, it is necessary to newly store the instruction set of Triple DES for the command register shown in previous drawing 29 . This will cause complication of the processing section constituted in a storage device 400.

[0464] Then, the data processor of this invention can perform processing corresponding to Triple DES encryption processing by considering logic which the cipher-processing section 401 by the side of a storage device 400 has as a single DES configuration, and proposes the configuration which made it possible to store the encryption data (a key, contents, etc.) based on a Triple DES method in the external memory 402 of a storage device.

[0465] For example, in the example of the data format type 0 shown in drawing 32 , in case download of contents data is performed from the record regenerator 300 to a storage device 400, authentication processing is performed at step S101 of drawing 39 which shows the flow of download of the format type 0 explained previously, and the session key Kses is generated here. Furthermore, in step S117, encryption processing of the contents key Kcon with the session key Kses is performed in the cipher-processing section 302 by the side of the record regenerator 300. This encryption key is transmitted to a storage device 400 through means of communications, and it sets to step S118. The cipher-processing section 403 of a storage device 400 which received this encryption key performs decode processing of the contents key Kcon with the session key Kses, and encryption processing of the contents key Kcon with the preservation key Kstr is performed further. Transmit this to the cipher-processing section 302 of the record regenerator 300, and the data with which the record regenerator 300 formed the data format (step S121), and formatting was carried out are transmitted to a storage device 400 after that. Processing which stores in external memory 402 the data which the storage device 400 received is performed.

[0466] If cipher processing in the cipher-processing section 401 of the storage device 400 performed between step S117 and S118 in the above-mentioned processing is alternatively considered as the configuration which can be performed, the method of Single DES or one of Triple DES In any case, correspondence becomes possible, also

when a contents offer contractor offers the contents data using the contents key Kcon according to Triple DES, and also when offering the contents data using the contents key Kcon according to Single DES.

[0467] The flow explaining the configuration which performs the code art which followed the Triple DES method at drawing 56 using the both sides of the cipher-processing section 302 of the record regenerator 300 in the data processor of this invention and the cipher-processing section 401 of a storage device 400 is shown. By drawing 56 , it is the example of encryption processing of the contents key Kcon using the preservation key Kstr performed in case contents data are downloaded from the record regenerator 300 to a storage device 400 as an example, and the example in case the contents key Kcon is a key by the Triple DES method is shown. In addition, although the example of processing is shown on behalf of the contents key Kcon, processing with the same said of other data is possible for other keys or contents here.

[0468] In a Triple DES method, as explained in drawing 8 -10 of the point, when based on a 64-bit key and a Triple DES method, in Single DES, it is the processing for which two or three keys are used as 128 bits or a 192-bit key configuration. These three contents keys are set to Kcon1, Kcon2, and (Kcon3), respectively. Since Kcon3 may not be used, the parenthesis shows.

[0469] Processing of drawing 56 is explained. Step S301 is the record regenerator 300 and a mutual recognition processing step between storage devices 400. This mutual recognition processing step is performed by processing of drawing 20 explained previously. In addition, the session key Kses is generated in the case of this authentication processing.

[0470] Termination of authentication processing of step S301 performs each check value, the check value A, the check value B, a contents check value, the total check value, and each collating processing of ICV in step S302.

[0471] These check value (ICV) collating processings are completed, and if judged with there being no data alteration, will progress to step S303 and it will set to the record regenerator 300. The control section 306 of the record regenerator cipher-processing section 302 The media 500 which received using the delivery key Kdis which took out previously or was generated using the code / decryption section 308 of the record regenerator cipher-processing section 302, Or decryption processing of the contents key Kcon stored in the header unit of data which received from means of communications 600 through the communications department 305 is performed. The contents key in this case is a key by the Triple DES method, and are

the contents keys Kcon1 and Kcon2 and (Kcon3).

[0472] Next, in step S304, the control section 306 of the record regenerator cipher-processing section 302 enciphers in the code / decryption section 308 of the record regenerator cipher-processing section 302 with the contents keys Kcon1 and Kcon2 decrypted at step S303, and the session key Kses which shared only the contents key Kcon1 in (Kcon3) on the occasion of mutual recognition.

[0473] The control section 301 of the record regenerator 300 reads the data containing the contents key Kcon1 enciphered with the session key Kses from the record regenerator cipher-processing section 302 of the record regenerator 300, and transmits these data to a storage device 400 through the record device controller 303 of the record regenerator 300.

[0474] Next, in step S305, the storage device 400 which received the contents key Kcon1 transmitted from the record regenerator 300 decrypts the received contents key Kcon1 with the session key Kses shared on the occasion of mutual recognition in the code / decryption section 406 of the storage device cipher-processing section 401. Furthermore, in step S306, it is made to re-encipher with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401, and transmits to the record regenerator 300 through the communications department 404.

[0475] Next, in step S307, the control section 306 of the record regenerator cipher-processing section 302 enciphers in the code / decryption section 308 of the record regenerator cipher-processing section 302 with the contents keys Kcon1 and Kcon2 decrypted at step S303, and the session key Kses which shared only the contents key Kcon2 in (Kcon3) on the occasion of mutual recognition.

[0476] The control section 301 of the record regenerator 300 reads the data containing the contents key Kcon2 enciphered with the session key Kses from the record regenerator cipher-processing section 302 of the record regenerator 300, and transmits these data to a storage device 400 through the record device controller 303 of the record regenerator 300.

[0477] Next, in step S308, the storage device 400 which received the contents key Kcon2 transmitted from the record regenerator 300 decrypts the received contents key Kcon2 with the session key Kses shared on the occasion of mutual recognition in the code / decryption section 406 of the storage device cipher-processing section 401. Furthermore, in step S309, it is made to re-encipher with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401, and transmits to the record regenerator 300

through the communications department 404.

[0478] Next, in step S310, the control section 306 of the record regenerator cipher-processing section 302 enciphers in the code / decryption section 308 of the record regenerator cipher-processing section 302 with the contents keys Kcon1 and Kcon2 decrypted at step S303, and the session key Kses which shared only the contents key Kcon3 in (Kcon3) on the occasion of mutual recognition.

[0479] The control section 301 of the record regenerator 300 reads the data containing the contents key Kcon3 enciphered with the session key Kses from the record regenerator cipher-processing section 302 of the record regenerator 300, and transmits these data to a storage device 400 through the record device controller 303 of the record regenerator 300.

[0480] Next, in step S311, the storage device 400 which received the contents key Kcon3 transmitted from the record regenerator 300 decrypts the received contents key Kcon3 with the session key Kses shared on the occasion of mutual recognition in the code / decryption section 406 of the storage device cipher-processing section 401. Furthermore, in step S312, it is made to re-encipher with the preservation key Kstr of a storage device proper saved at the internal memory 405 of the storage device cipher-processing section 401, and transmits to the record regenerator 300 through the communications department 404.

[0481] Next, in step S313, the cipher-processing section of a record regenerator forms various kinds of data formats explained by drawing 32 -35, and transmits to a storage device 400.

[0482] Finally in step S314, a storage device 400 stores in external memory 402 the received data which format formation ended. The contents keys Kcon1 and Kcon2 and (Kcon3) which were enciphered with the preservation key Kstr are included in this format data.

[0483] By performing such processing, it becomes possible to store the contents key stored in a storage device 400 as a key by the cipher system of a Triple DES method. In addition, when contents keys are two keys, Kcon1 and Kcon2, processing of steps S310-S312 is omitted.

[0484] Thus, a storage device 400 becomes storable in memory about the key with which Triple DES was applied by performing processing of the same mode, i.e., the processing step of steps S305 and S306, [change / multiple times and its object]. What is necessary is to perform steps S305 and S306, to perform formatting processing of step S313, and just to store in memory, when the contents key Kcon is an application key of Single DES. What is necessary is to store such a configuration in

the command register of drawing 29 which explained previously the command which performs processing of steps S305 and S306, and just to consider it as the configuration which performs this processing once to 3 times suitably with a single DES method, the mode, i.e., the Triple DES method, of a contents key. Therefore, processing of the both sides of Triple DES method and single DES method ** is attained, without including mode of processing of Triple DES in the processing logic of a storage device 400. In addition, about a cipher system, it is possible to judge by recording on the handling plan in the header unit of contents data, and referring to this.

[0485] (14) A contents type and starting priority information are included in the handling plan stored in the header unit of the contents data used in the data processor of this invention so that I may be understood from the contents data configuration of drawing 4 -6 explained to the program starting processing place based on the starting priority in the handling plan in contents data. The record regenerator 300 in the data processor of this invention determines the starting ranking of these contents according to starting priority information, when two or more a storage device 400 or DVDs and CDs, hard disks, and accessible contents data further recorded on various record media, such as a game cartridge, exist.

[0486] The record regenerator 300 gives priority to and performs the program in contents data with the highest priority according to the priority information in [after performing authentication processing with various storage devices, such as each storage device DVD equipment, CD drive equipment, and hard disk drive equipment,] contents data. Hereafter, this "program starting processing based on the starting priority in the handling plan in contents data" is explained.

[0487] It set to explanation of the data-processor example of this invention mentioned above, and processing in case the record regenerator 300 reproduces and performs contents data from one storage device 400 was explained as a core. However, generally, the record regenerator 300 has a configuration accessible to various record media, such as DVD, CD, a hard disk, a memory card further connected through PIO111 and SIO112, and a game cartridge, other than a storage device 400 through the reading section 304, as shown in drawing 2 . In addition, although the one reading section 304 is indicated by drawing 2 in order to avoid complication of drawing, the record regenerator 300 can equip juxtaposition with a different storage, for example, DVD and CD, a floppy disk, and a hard disk.

[0488] The record regenerator 300 is accessible to two or more storages, and contents data are stored in each storage, respectively. For example, the contents data which the content provider of the exteriors, such as CD, supplies are stored in

each storage, such as a memory card, with the contents data configuration of drawing 26 and drawing 27 , when it is stored in media with the data configuration of above-mentioned drawing 4 and downloads through these media or means of communications. Furthermore, it is stored in a format which is specifically different on media and a storage device, respectively as shown in drawing 32 -35 according to the format type of contents data. However, in any case, a contents type and starting priority information are included at the handling plan in the header of contents data.

[0489] Contents starting processing of a record regenerator when access to the contents data of these plurality is possible is explained according to a flow.

[0490] Drawing 57 is a processing flow which shows the example of processing (1) in case there are two or more contents which can be started. Step S611 is a step which performs authentication processing of a storage device with the accessible record regenerator 300. A memory card, DVD equipment, CD drive, a hard disk drive unit, the game cartridge connected through PIO111 and SIO112 are further contained in an accessible storage device. Authentication processing is performed according to the procedure previously explained to the basis of the control of a control section 301 shown by drawing 2 by drawing 20 as opposed to each storage device.

[0491] Next, it sets step S612 and the program which can be started from the contents data stored in the memory in the storage device which succeeded in authentication is detected. Specifically, this is performed as processing which extracts that whose contents type included in the handling plan of contents data is a program.

[0492] Next, in step S613, the starting priority in the program which was extracted at step S612 and which can be started is judged. This is processing which compares the priority information specifically included in the handling information in the header of the contents data which were chosen in step S612, and in which two or more startings are possible, and chooses the highest priority.

[0493] Next, the program chosen at step S614 is started. In addition, when the priority set up in the program in which two or more startings are possible is the same, the contents program which set up priority default between storage devices and was stored in the device given top priority is performed.

[0494] The identifier was set as two or more storage devices, and the example of processing (2) in case there is two or more processing mode which performs authentication processing and a contents program search, i.e., the contents which can be started, was shown in drawing 58 one by one about the storage device to which each identifier was given.

[0495] It is the step which performs authentication processing (refer to drawing 20) of the storage device (i) with which the record regenerator 300 was equipped at step S621. The identifier of 1 – n is given to the storage device of plurality (n pieces) one by one.

[0496] When it judges whether authentication at step S621 was successful and authentication is successful, it progresses to step S623 and the program which can be started is searched with step S622 out of the record medium of the storage device (i). When authentication is not successful, it progresses to step S627 and the existence of the storage device in which new contents retrieval is possible is judged, when there is nothing, processing is ended, when a storage device exists, it progresses to step S628, the storage device identifier i is updated, and the authentication processing step after step S621 is repeated.

[0497] The processing in step S623 is processing which detects the program which can be started from the contents data stored in the storage device (i). Specifically, this is performed as processing which extracts that whose contents type included in the handling plan of contents data is a program.

[0498] At step S624, when it is judged and extracted whether that whose contents type is a program was extracted, in step S625, what has the highest priority is chosen among an extract program, and a selection program is performed in step S626.

[0499] the case where it is judged with that whose contents type is a program not having been extracted in step S624 -- step S627 -- progressing -- new contents -- the existence of a storage device [****] is judged, when there is nothing, processing is ended, when a storage device exists, it progresses to step S628, the storage device identifier i is updated, and the authentication processing step after step S621 is repeated.

[0500] Drawing 59 is a processing flow which shows the example of processing (3) in case there are two or more contents which can be started. Step S651 is a step which performs authentication processing of a storage device with the accessible record regenerator 300. Authentication processing of accessible DVD equipment, CD drive, a hard disk drive unit, a memory card, a game cartridge, etc. is performed. Authentication processing is performed according to the procedure previously explained to the basis of the control of a control section 301 shown by drawing 2 by drawing 20 as opposed to each storage device.

[0501] Next, it sets step S652 and the program which can be started from the contents data stored in the memory in the storage device which succeeded in authentication is detected. Specifically, this is performed as processing which

extracts that whose contents type included in the handling plan of contents data is a program.

[0502] Next, in step S653, information, such as a name of the program which was extracted at step S652 and which can be started, is displayed on a display means. In addition, although the display means is not shown by drawing 2 , it has composition outputted to the display means which the data outputted as AV output data do not illustrate. In addition, user provided information, such as a program name of each contents data, is stored in the identification information of contents data, and the program name of each contents data [finishing / authentication] etc. outputs program information to an output means through a control section 301 at the basis of control of Maine CPU 106 shown in drawing 2 .

[0503] Next, at step S654, Maine CPU 106 receives the program selection input by the user from input means, such as an input interface shown in drawing 2 , a controller, a mouse, and a keyboard, through the input interface 110, and a user selection program is performed in step S655 according to a selection input.

[0504] Thus, in the data processor of this invention, since it considered as the configuration which displays bootstrap information on a display means and is chosen by the user or it stored program starting priority information in the handling information in the header in contents data and the record regenerator 300 started the program according to this priority, a user does not need to search a program, and it becomes possible to exclude the effort of the time amount and the user who require for starting. Moreover, the complicated nature of processings, like since the display of being the program which can be started [starting or] is made after authentication processing of a storage device, all the programs that can be started check justification, after choosing a program is canceled.

[0505] (15) In the data processor of a contents configuration and playback (expanding) processing this invention, as mentioned above, the record regenerator 300 performs download or a storage device 400 to regeneration for contents from media 500 or means of communications 600. The above-mentioned explanation has explained processing of the encryption data accompanying download of contents or regeneration as a core.

[0506] The control section 301 in the record regenerator 300 of drawing 3 controls decryption processing download processing of the devices 500, such as DVD which offers contents data, means of communications 600, and the contents data from a storage device or the authentication processing accompanying regeneration, encryption, and at large.

[0507] The refreshable contents obtained as these processing results are voice data, image data, etc. Decode data are put under control of Maine CPU shown in drawing 2 from a control section 301, and are outputted to AV output section according to voice data, image data, etc. However, contents are voice data, and if MP3 compression is made, decode processing of voice data will be made and outputted by the MP3 decoder of AV output section shown in drawing 2 . Moreover, expanding processing will be performed and outputted by the MPEG 2 decoder of AV processing section, if contents data are image data and are an MPEG 2 compression image. Thus, compression (coding) processing may be made, and the data contained in contents data also have data with which compression processing is not performed, and perform and output the processing according to contents.

[0508] However, there are various classes of compression processing and expanding processing programs, and when there is no expanding processing executive program which corresponds even if compressed data is offered from a content provider, the situation where this is unreproducible occurs.

[0509] Then, the data processor of this invention indicates the configuration which stores compressed data and its decode (expanding) processing program collectively in data contents, or the configuration which stores the link information of compressed data and a decode (expanding) processing program as header information of contents data.

[0510] From the data-processing general drawing shown in drawing 2 , drawing into which the element and the related element about this configuration were packed briefly is shown in drawing 60 . The record regenerator 300 receives offer of various contents from the storage devices 400, such as a memory card which stored the devices 500, such as DVD and CD, means of communications 600, or contents. Various data, such as that to which these contents are voice data, a static image, dynamic-image data, program data, etc., and encryption processing is performed, a thing which is not given and a thing by which compression processing is made, and a thing which is not made, are contained.

[0511] When receipt contents are enciphered, decode processing is performed by control of a control section 301, and cipher processing of the cipher-processing section 302 by technique which was explained in the item already mentioned above. After the decoded data are transmitted to AV processing section under control of Maine CPU 106 109 and being stored in the memory 3090 of AV processing section 109, analysis of a contents configuration is performed in the contents analysis section 3091. For example, if the data decompression program is stored in contents, a

program is stored in the program store section 3093, and if data, such as voice data and image data, are contained, these will be memorized in the data storage section 3092. In the expanding processing section 3094, expanding processing of the compressed data which was memorized by the program store section and which was memorized by the data storage section 3092, for example using expanding processing programs, such as MP3, is performed, and it is outputted to a loudspeaker 3001 and a monitor 3002.

[0512] Next, some examples of the configuration of the data which AV processing section 109 receives through a control section 301, and processing are explained. In addition, although explained on behalf of what showed voice data as an example of contents, and applied MP3 as an example of a compression program here, it can apply not only to voice data but to image data, and this configuration can apply not only MP3 but MPEG 2 and the program of 4 grade various kinds also about a compression expanding processing program.

[0513] The example of a contents configuration is shown in drawing 61 . Drawing 61 is the music data 6102 compressed by MP3 and the example which combined the MP3 decode (expanding) processing program 6101, and was constituted as one contents. These contents are stored in media 500 or a storage device 400 as 1 contents, or are distributed from means of communications 600. After it will perform decode processing by the cipher-processing section 303 if enciphered as these contents explained the record regenerator 300 previously, it is transmitted to AV processing section 109.

[0514] In the contents analysis section 3091 of AV processing section 109, the received contents are analyzed, from the contents which consist of the voice data expanding program (MP3 decoder) section and the compression voice data section, the voice data expanding program (MP3 decoder) section is taken out, a program is memorized in the program store section 3093, and compression voice data is memorized in the data storage section 3092. In addition, the contents analysis section 3091 may receive information received independently, such as a contents name and contents configuration information, or may perform contents analysis based on the data to which discernment data, such as a data name included in contents, a data length, a data configuration, etc. are indicated to be contents. Next, the compression expanding processing section 3094 performs expanding processing of the MP3 compression voice data memorized by the data storage section 3092 according to the voice data expanding program (MP3 decoder) memorized by the program store section 3093, and AV processing section 109 outputs the elongated voice data to a

loudspeaker 3001.

[0515] The flow which shows an example of regeneration of the data which have the contents configuration of drawing 61 in drawing 62 is shown. If step S671 is the contents of the data name stored in the memory 3090 of AV processing section 109, for example, music data, it will be taken out from the data in the information which received information, such as a music name, apart from contents, or contents, and will be displayed on a monitor 3002. Step S672 receives a user's selection through the input interface 110 from various input means, such as a switch and a keyboard, and outputs the regeneration instruction based on user input data to the basis of control of CPU106 at AV processing section 109. AV processing section 109 performs extract of the data based on an own alternative, and expanding processing in step S673.

[0516] Next, the example of a configuration by which either compression voice data or an expanding processing program is contained in one contents, and the contents information which shows the contents of contents as header information of each contents further is included in drawing 63 is shown.

[0517] As shown in drawing 63 , when contents are programs 6202, the contents identification information which shows that it is a program as header information 6201 and that a program class is an MP3 expanding program is contained. On the other hand, when voice data 6204 is included as contents, the information that it is MP3 compressed data is included in the contents information on a header 6203. It adds to the contents which choose only information required for playback from the data contained in the handling plan (refer to drawing 5) of the contents data configuration which was mentioned above, and which is shown, for example in drawing 4 , and are transmitted to AV processing section 109, and this header information can be constituted. The discernment value of the handling plan data which are specifically needed for each configuration data in the "handling plan" shown in drawing 5 in the cipher-processing section 302, and the data which are needed at the time of the regeneration in AV processing section 109 is added, and these discernment value can extract only what shows a required thing in AV processing section 109, and can consider as header information.

[0518] According to header information, the contents analysis section 3091 of AV processing section 109 which received each contents shown in drawing 63 memorizes program contents in the program store section 3093, when it is a program, and when it is data, it memorizes data contents in the data storage section 3092. Then, the compression expanding processing section 3094 takes out data from the data storage

section, and performs and outputs expanding processing according to the MP3 program memorized in the program store section 3093. In addition, when the same program is already stored in the program store section 3093, program storing processing may be omitted.

[0519] The flow which shows an example of regeneration of the data which have the contents configuration of drawing 63 in drawing 64 is shown. If step S675 is the contents of the data name stored in the memory 3090 of AV processing section 109, for example, music data, it will be taken out from the header in the information which received information, such as a music name, apart from contents, or contents, and will be displayed on a monitor 3002. Step S676 receives a user's selection through the input interface 110 from various input means, such as a switch and a keyboard.

[0520] Next, the program for playback of the data corresponding to an own alternative (for example, MP3) is searched with step S677. It is desirable that this candidate for program retrieval makes the accessible range of the record playback device 300 the maximum retrieval range, for example, it also makes the retrieval range each media 500 and means of communications 600 which are shown in drawing 60 , and storage device 400 grade.

[0521] The contents passed to AV processing section 109 are only data division, and program contents may be stored in other record media in the record regenerator 300, and may be offered by the contents offer contractor through media, such as DVD and CD. therefore, the candidate for retrieval -- access of the record playback device 300 -- let the range [****] be the retrieval range. If a playback program is found as a result of retrieval, the regeneration instruction based on user input data will be outputted to the basis of control of CPU106 at AV processing section 109. AV processing section 109 performs extract of the data based on an own alternative, and expanding processing in step S679. Moreover, a program is searched before step S675 and you may make it display only the data with which the program was detected in step S675 as another example.

[0522] Next, the example of a configuration by which the compression voice data 6303 and the expanding processing program 6302 are contained in one contents, and the playback priority information on contents is further included in drawing 65 as header information 6301 of contents is shown. This is the example which added playback priority information to the contents configuration of previous drawing 61 as header information. This determines the order of playback based on the playback priority set up between the contents which AV processing section 109 received like the above-mentioned "program starting processing based on the starting priority in the

handling plan in (14) contents data."

[0523] The flow which shows an example of regeneration of the data which have the contents configuration of drawing 65 in drawing 66 is shown. Step S681 sets the data information of the data stored in the memory 3090 of AV processing section 109, i.e., the data for playback, as a retrieval list. a retrieval list -- a part of memory in AV processing section 109 -- it sets up using a field. Next, in step S682, regeneration of the data which chose the high data of priority and were chosen from the retrieval list in step S683 in the contents analysis section 3091 of AV processing section 109 is performed.

[0524] Next, the example of a configuration by which playback priority information is added only to the header 6403 of data contents at drawing 67 in the example which consists of header information, the program data 6402 or header information 6403, and one combination of the compressed data 6404 at one contents is shown.

[0525] The flow which shows an example of regeneration of the data which have the contents configuration of drawing 67 in drawing 68 is shown. Step S691 sets the data information of the data stored in the memory 3090 of AV processing section 109, i.e., the data for playback, as a retrieval list. a retrieval list -- a part of memory in AV processing section 109 -- it sets up using a field. Next, in step S692, the high data of priority are chosen from a retrieval list in the contents analysis section 3091 of AV processing section 109.

[0526] Next, the program for data playback corresponding to selected data (for example, MP3) is searched with step S693. processing [in / in this candidate for program retrieval / the flow of previous drawing 64] -- the same -- access of the record playback device 300 -- it is desirable to make the range [****] into the maximum retrieval range, for example, it also makes the retrieval range each media 500 shown in drawing 60 , means of communications 600, and storage device 400 grade.

[0527] as the result of retrieval -- a playback program -- being found (it being Yes at step S694) -- in step S695, expanding regeneration is performed using the program acquired as a result of retrieval of selected data.

[0528] On the other hand, when a program is not detected as a retrieval result (it is Yes at step S694), it progresses to step S696 and the regeneration using the same program deletes a required thing in other data contained during the retrieval list set up at step S691. This is because it is clear not to be detected even if it newly performs a playback program search to the data. Furthermore, when it is not because it judges whether a retrieval list is empty in step S697, return and data with the following, still

higher priority are extracted to step S692, and program retrieval processing is performed.

[0529] According to this configuration, thus, the contents by which compression processing was carried out When it is only the data with which it was constituted with the decode (expanding) BUROGURAMU, or contents were compressed, or an expanding processing program Since it has the header information which shows what kind of processing [that contents are what kind of compressed data or] is performed to each contents The processing section (for example, AV processing section) which received contents Perform expanding regeneration using the expanding processing program attached to compressed data, or an expanding processing program is searched based on the header information of compressed data. Since expanding regeneration is performed according to the program acquired as a result of retrieval, processing of selection of the expanding program of the data by the user, retrieval, etc. becomes unnecessary, a user burden is mitigated and efficient data playback is attained. Furthermore, the configuration which sets playback sequence automatically to a header according to the configuration with playback priority information is attained, and actuation of the order setup of playback by the user can be omitted.

[0530] In addition, in the above-mentioned example, although MP3 as compression voice data contents and an expanding processing program of speech compression data was explained as an example, even if it is the contents containing compressed data, and the contents which have the expanding processing program of compression image data, this configuration can be applied similarly and does the same effectiveness so.

[0531] (16) Generation of save data and storing in a storage device, and the data processor of regeneration this invention interrupt a game program on the way, when the contents performed in the record regenerator 300 are game programs etc., and they have the configuration which the game condition at the interruption time etc. is stored in save, i.e., a storage device, and this is read at the time of a restart, and can be continued a game after predetermined time to newly resume.

[0532] Although the save data storage configuration in record regenerators, such as the conventional game device and a personal computer, has the configuration which saves save data at storages, such as built-in or a memory card in which external is possible, a floppy disk, a game cartridge, or a hard disk, for example, in a record regenerator, it has the composition that do not have a security secured configuration to the save data, for example, save processing of data is especially performed by the specification common to a game application program.

[0533] The situation which the save data which followed, for example, were saved using one certain record regenerator A are used by another game program, or is rewritten occurs, and the actual condition is that most security of save data was not taken into consideration conventionally.

[0534] The data processor of this invention offers the configuration which made security reservation of such save data realizable. For example, only the game program enciphers based on usable information, and the save data of a certain game program are stored in a storage device. Or it enciphers based on the information on a record regenerator proper, and stores in a storage device. By such technique, use of save data can be restricted only to a specific device and a specific program, and the security of save data is secured. Hereafter, "the generation of save data and storing in a storage device, and regeneration" in the data processor of this invention are explained.

[0535] The block diagram which explains the save data storage processing in the data processor of this invention to drawing 69 is shown. The record regenerator 300 is provided with contents from the media 500, such as DVD and CD, or means of communications 600. The contents offered are enciphered with the contents key Kcon which is a key of a contents proper, as explained previously, and after the record regenerator 300 acquires a contents key according to the processing explained in the column of "download processing to a storage device from (7) record regenerator" mentioned above (refer to drawing 22) and decodes encryption contents, it is stored in a storage device 400. Here, the record regenerator 300 decodes a contents program from media and means of communications. Playback, After activation the save data which perform and are obtained External or a built-in memory card, It stores in either of various kinds of storage devices 400A, 400B, and 400C, such as a hard disk. After downloading the processing to reproduce or contents to storage device 400A, contents are reproduced and performed from storage device 400A. The save data is stored in the processing storage device 400 stored in either of various kinds of storage devices 400A, 400B, and 400C, such as external or a built-in memory card, and a hard disk, and the processing to reproduce is explained.

[0536] In the record regenerator 300, it has the master key which generates the record regenerator signature key Kdev which is a signature key of a proper, and further various kinds of individual keys to the system signature key Ksys which is a signature key common to the record regenerator identifier IDdev and a system as explained previously, and each record regenerator. It is the key which generates the delivery key Kdis or the authentication key Kake as the master key was explained in

detail in "the cipher-processing key generation configuration based on (12) master keys." It is shown as MKx especially here as a thing representing the master key at large which the record regenerator 300 has, without limiting the class of master key. The example of the cryptographic key Ksav of save data was shown in the lower berth of drawing 69 . The save data cryptographic key Ksav is a cryptographic key used for the decode processing at the time of reproducing save data from the encryption processing in the case of storing in various storage device 400 A-C, and various storage device 400 A-C. The example of storing processing of save data and regeneration is explained below using drawing 70 .

[0537] Drawing 70 is the flow Fig. of the processing which stores save data in one of storage device 400 A-C using contents ****** or a system common key. In addition, the processing in each flow is processing which the record regenerator 300 performs, and the storage device which stores save data by each flow is not a limited **** thing at either that what is necessary is just built-in or external storage device 400 A-C.

[0538] Step S701 is processing whose record regenerator 300 reads the contents identifier ID, for example, a game. This is data contained in the identification information in drawing 4 explained previously and the contents data shown in 26, 27, 32-35, and Maine CPU 106 which received the storing processing instruction of save data through the input interface 110 shown in drawing 2 directs reading of a contents identifier to a control section 301.

[0539] When control sections 301 are the contents by which the identification information contained in the header in contents data through a read station 304 was taken out in the case of the contents by which the executive program is performed through the read stations 304, such as DVD and CD-ROM, and the executive program was stored in the storage device 400, identification information is taken out through the record device controller 303. In addition, the record regenerator 300 is performing a contents program, and when a contents identifier is storing ending, the identification information contained in read data may already be used for RAM in a record regenerator, and other accessible record media, without performing new reading processing.

[0540] Next, step S702 is a step which changes processing by whether a use limit of a program is performed. It is the limit information set up whether a program use limit is ** attached in the limit which makes available the save data to save at a proper only at the program, and when carrying out that it is available in a proper only to a program, the case where carry out as "those with a program use limit", and it carries out as the save data which do not have use restrained by the program carries out as "with no

program use limit." A user may enable it to set this as arbitration, a contents manufacturer may set up, this information may be stored in a contents program, and the set-up limit information is stored in storage device 400 A-C of drawing 69 as a data control file.

[0541] The example of a data control file is shown in drawing 71 . A data control file is generated as a table which includes a data number, a contents identifier, a record regenerator identifier, and a program use limit as an item. A contents identifier is discernment data of the contents program used as the object which stores save data. A record regenerator identifier is [IDdev] shown in the identifier of a record regenerator which stored save data, for example, drawing 69 . A program use limit serves as a setup of "not carrying out", when considering at a proper the save data saved as mentioned above as a setup of "carrying out" a ** case only at the program for it to be available and enabling use which is not restricted to a correspondence program. The user using a contents program may enable it to set a program use limit as arbitration, and a contents manufacturer may set it up, and it may store this information in a contents program.

[0542] Explanation of return and a flow is continued to drawing 70 . In SUTEBBU S702, when a setup of "carrying out" is carried out about the program use limit, it progresses to step S703. At step S703, the key Kcon of a contents proper, for example, the contents key explained previously, is read from contents data, and a contents proper key is made into the save data cryptographic key Ksav, or the save data cryptographic key Ksav is generated based on a contents proper key.

[0543] On the other hand, in SUTEBBU S702, when a setup of "not carrying out" is carried out about the program use limit, it progresses to step S707. At step S707, the system common key Ksys stored in the record regenerator 300, for example, a system signature key, is read from the internal memory 307 of the record regenerator 300, and the system signature key Ksys is made into the save data cryptographic key Ksav, or the save data cryptographic key Ksav is generated based on a system signature key. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0544] Next, it sets to step S704 and save data encryption processing is performed using the save data encryption key Ksav chosen or generated at step S703 or step S707. The cipher-processing section 302 in drawing 2 performs this encryption processing with the application of the above-mentioned DES algorithm.

[0545] The save data by which encryption processing was carried out in step S704 are

stored in a storage device in step S705. As the storage device which can store save data shows drawing 69 , when there are more than one, a user chooses beforehand either of storage device 400 A-C as the save data storage point. Furthermore, the writing of the writing of the program use limit information previously set as the data control file previously explained using drawing 71 in step S706 at step S702, i.e., a program use limit, "it carries out", and "not carrying out" is performed.

[0546] Above, storing processing of save data is completed. Yes, i.e., selection of "carrying out a program use limit", is made in step S702, decode processing the save data by which encryption processing was carried out with the save data-encryption key Ksav generated based on the contents proper key in step S703 according to a contents program without contents proper key information becomes impossible, and save data can use only in the contents program which it has in the same contents key information. However, since the save data encryption key Ksav was not generated based on the information on a record regenerator proper, the save data stored in removable storage devices, such as a memory card, for example become refreshable here, as long as it uses it with the contents program which corresponds also in a different record regenerator.

[0547] Moreover, in step S702, No, i.e., selection of "not carrying out a program use limit", is made, and the save data by which encryption processing was carried out in step S707 with the save data encryption key Ksav based on a system common key become that reproducing and using is possible, even when the program from which a contents identifier differs is used, and even when record regenerators differ.

[0548] Drawing 72 is the flow which showed the processing which reproduces the save data stored by save data storage processing of drawing 70 .

[0549] Step S711 is processing whose record regenerator 300 reads the contents identifier ID, for example, a game. This is the same processing as step S701 of save data storage processing of drawing 70 explained previously, and is processing which reads the data contained in the identification information in contents data.

[0550] Next, at step S712, the contents identifier which read the data control file explained using drawing 71 , and was read from storage device 400 A-C shown in drawing 69 in step S711, and the use program limit information set up by corresponding are extracted. When the program use limit set as the data control file is "carrying out", it progresses to step S714, and when it is "not carrying out", it progresses to step S717.

[0551] At step S714, the key Kcon of a contents proper, for example, the contents key explained previously, is read from contents data, and a contents proper key is

used as the save data decryption key Ksav, or the save data decryption key Ksav is generated based on a contents proper key. The decryption key generation algorithm used as that to which the processing algorithm corresponding to encryption key generation processing in this decryption key generation processing can decode the data which were applied and were enciphered based on a certain contents proper key with the decode key generated based on the same contents proper key is applied.

[0552] On the other hand, in SUTEBBU S712, when a setup of a data control file is a setup of "not carrying out" about a program use limit In step S717, the system common key Ksys stored in the record regenerator 300, for example, a system signature key, is read from the internal memory 307 of the record regenerator 300. The system signature key Ksys is used as the save data decryption key Ksav, or the save data decryption key Ksav is generated based on a system signature key. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0553] Next, it sets to step S715, decryption processing of save data is performed using the save data decryption key Ksav chosen or generated at step S714 or step S717, and the decoded save data are set and performed [reproduce and] to the record regenerator 300 in step S716.

[0554] Above, regeneration of save data is completed. When a setup of "carrying out a program use limit" is made by the data control file as mentioned above, a save data decryption key is generated based on a contents proper key, and when a setup of "not carrying out a program use limit" is, a save data decryption key is generated based on a system common key. When a setup of "carrying out a program use limit" is carried out, if the contents identifier of the contents currently used is not the same, the possible decryption key of decode processing of save data can be obtained, and it becomes possible to raise the security of save data.

[0555] Drawing 73 and drawing 74 are the save data storage processing flows (drawing 73) and save data regeneration flows (drawing 74) which generate a save data encryption key and a decryption key using a contents identifier.

[0556] In drawing 73 , steps S721-S722 are the same processings as steps S701-S702 of drawing 70 , and omit explanation.

[0557] When "a program use limit is carried out" is set up in step S722, the save data storage processing flow of drawing 73 reads a contents identifier, i.e., content ID, from contents data in step S723, and uses content ID as the save data encryption key Ksav, or generates the save data encryption key Ksav based on content ID. For example,

the cipher-processing section 307 of the record regenerator 300 can apply the master key MKx stored in the internal memory of the record regenerator 300 to the content ID read from contents data, for example, can obtain the save data encryption key Ksav by DES (MKx, content ID). Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0558] On the other hand, when it considers as a setup of "not carrying out" about a program use limit in SUTEBBU S722, in step S727, the system common key Ksys stored in the record regenerator 300, for example, a system signature key, is read from the internal memory 307 of the record regenerator 300, and the system signature key Ksys is used as the save data encryption key Ksav, or the save data encryption key Ksav is generated based on a system signature key. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0559] Step S 724 or less processing is the same as the processing not more than step S704 in the processing flow of above-mentioned drawing 70, and omits explanation.

[0560] Furthermore, drawing 74 is a processing flow which reproduces the save data stored in the storage device by the save data storage processing flow of drawing 73 , and is performed, and that of steps S731-S733 is the same as that of correspondence processing of above-mentioned drawing 72 , and only steps S734 differ. In step S734, a contents identifier, i.e., content ID, is read from contents data, and content ID is used as the save data decryption key Ksav, or the save data decryption key Ksav is generated based on content ID. The decryption key generation algorithm used as that to which the processing algorithm corresponding to encryption key generation processing in this decryption key generation processing can decode the data which were applied and were enciphered based on a certain contents identifier with the decode key generated based on the same contents identifier is applied.

[0561] Since the following processings and steps S735, S736, and S737 are the same as that of correspondence processing of drawing 72 , explanation is omitted. Except when a corresponding contents program has consistency like [since it considered as the configuration which generates a save data-encryption key and a decryption key using content ID when having followed drawing 73 , the save data storage of drawing 74 , and regeneration and a carrying-out-program use limit setup was performed] the save data storage and the regeneration which used the previous contents proper key, it becomes with the configuration that save data cannot use, and it becomes that it is

possible in the preservation raised save data security.

[0562] Drawing 75 and drawing 77 are the save data storage processing flows (drawing 75) and save data regeneration flows (drawing 77) which generate a save data encryption key and a decryption key using a record regenerator proper key.

[0563] In drawing 75 , step S741 is the same processing as step S701 of drawing 70 , and omits explanation. Step S742 is a step which sets up whether a record regenerator is restricted or it does not carry out. When limiting an available record regenerator for save data, a record regenerator limit sets up the case where it is supposed that it is available only to the record regenerator which generated and stored save data, saying "It carries out", and considers the case where it is supposed with other record regenerators that it is available as a setup of "not carrying out." If it sets up "a record regenerator limit is carried out" in step S742, it will progress to step S743, and if it sets up "it does not carry out", it will progress to step S747.

[0564] The example of a data control file is shown in drawing 76 . A data control file is generated as a table which includes a data number, a contents identifier, a record regenerator identifier, and a record regenerator limit as an item. A contents identifier is discernment data of the contents program used as the object which stores save data. A record regenerator identifier is [IDdev] shown in the identifier of a record regenerator which stored save data, for example, drawing 69 . When limiting an available record regenerator for save data, a record regenerator limit sets up the case where it is supposed that it is available only to the record regenerator which generated and stored save data, saying "It carries out", and considers the case where it is supposed with other record regenerators that it is available as a setup of "not carrying out." The user using a contents program may enable it to set record regenerator limit information as arbitration, and a contents manufacturer may set it up, and it may store this information in a contents program.

[0565] In the save data storage processing flow of drawing 75 When "a record regenerator limit is carried out" is set up in step S742, In step S743 From the record regenerator 300 to a record regenerator proper key For example, the record regenerator signature key Kdev is read from the internal memory 307 of the record regenerator 300, and the record regenerator signature key Kdev is used as the save data encryption key Ksav, or the save data encryption key Ksav is generated based on the record regenerator signature key Kdev. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0566] On the other hand, in SUTEBBU S742, when it considers as a setup of "not

carrying out" about a record regenerator limit In step S747, the system common key Ksys stored in the record regenerator 300, for example, a system signature key, is read from the internal memory 307 of the record regenerator 300. The system signature key Ksys is used as the save data encryption key Ksav, or the save data encryption key Ksav is generated based on a system signature key. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0567] Processing of steps S744 and S745 is the same as the correspondence processing in the processing flow of above-mentioned drawing 70 , and omits explanation.

[0568] the record regenerator limit information which the user set as the data control file (refer to drawing 76) at a contents identifier, a record regenerator identifier, and step 742 in step S746 -- "-- it carries out -- /don't carry out -- " -- it writes in.

[0569] Furthermore, drawing 77 is a processing flow which reproduces the save data stored in the storage device by the save data storage processing flow of drawing 75, and is performed, and step S751 reads a contents identifier like correspondence processing of above-mentioned drawing 72. Next, in step S752, the record regenerator identifier (IDdev) stored in the memory in the record regenerator 300 is read.

[0570] record regenerator limit information [finishing / the contents identifier from a data control file (refer to drawing 76), a record regenerator identifier, and a setup / in step S753] -- "-- it carries out -- /don't carry out -- " -- each information is read. In the entry whose contents identifier in a data control file corresponds, when it differs from the record regenerator identifier in which the record regenerator identifier of a table entry was read at step S752 when record regenerator limit information was set up for "carrying out", processing is ended.

[0571] Next, when a setup of a data control file is "carrying out a record regenerator limit" at step S754, it progresses to step S755, and when it is "not carrying out", it progresses to step S758.

[0572] In step S755, the record regenerator proper key Kdev from the record regenerator 300, for example, a record regenerator signature key, is read from the internal memory 307 of the record regenerator 300, and the record regenerator signature key Kdev is used as the save data decryption key Ksav, or the save data decryption key Ksav is generated based on the record regenerator signature key Kdev. The decryption key generation algorithm used as that to which the processing

algorithm corresponding to encryption key generation processing in this decryption key generation processing can decode the data which were applied and were enciphered based on a certain record regenerator proper key with the decode key generated based on the same record regenerator proper key is applied. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0573] On the other hand, the system common key Ksys stored in the record regenerator 300, for example, a system signature key, is read from the internal memory 307 of the record regenerator 300, and the system signature key Ksys is used as the save data decryption key Ksav in step S758, or the save data decryption key Ksav is generated based on a system signature key. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav. The following steps S756 and S757 are the same processings as the correspondence step of the above-mentioned save data regeneration flow.

[0574] According to the save data storage and the regeneration flow which are shown in drawing 75 and drawing 77, since encryption and decryption processing are performed with a record regenerator proper key, the save data with which selection of "carrying out a record regenerator limit" was made become that it is possible in decoding and using in a record regenerator with the same record regenerator ******, i.e., the same record regenerator.

[0575] Next, a record regenerator identifier is used for drawing 78 and drawing 79, and the processing flow which generates a save data encryption and a decryption key, and is stored and reproduced is shown.

[0576] Drawing 78 performs a save data encryption using a record regenerator identifier, and stores it in a storage device. Steps S761-S763 are the same processings as previous drawing 75. At step S764, the save data encryption key Ksav is generated using the record regenerator identifier (IDdev) read from the record regenerator. The master key MKx which applied IDdev as a save data encryption key Ksav, or was stored in the internal memory of the record regenerator 300 is applied, and the save data encryption key ksav is generated based on IDdev, such as obtaining the save data encryption key Ksav by DES (MKx, IDdev). Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0577] The following processing steps S765-S768 are the same as that of

correspondence processing of above-mentioned drawing 75, and omit explanation.

[0578] Drawing 79 is a processing flow which reproduces the save data stored in the storage device by processing of drawing 78, and is performed. Steps S771-S774 are the same as that of correspondence processing of above-mentioned drawing 77.

[0579] At step S775, the decryption key Ksav of save data is generated using the record regenerator identifier (IDdev) read from the record regenerator. The master key MKx which applied IDdev as a save data decryption key Ksav, or was stored in the internal memory of the record regenerator 300 is applied, and the save data decryption key Ksav is generated based on IDdev, such as obtaining the save data decryption key Ksav by DES (MKx, IDdev). The decryption key generation algorithm used as that to which the processing algorithm corresponding to encryption key generation processing in this decryption key generation processing can decode the data which were applied and were enciphered based on a certain record regenerator identifier with the decode key generated based on the same record regenerator identifier is applied. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0580] The following processing steps S776-S778 are the same as that of processing of the correspondence step of above-mentioned drawing 76.

[0581] According to the save data storage and the regeneration flow which are shown in this drawing 78 and drawing 79, since encryption and decryption processing are performed by the record regenerator identifier, the save data with which selection of "carrying out a record regenerator limit" was made become that it is possible in decoding and using in a record regenerator with the same record regenerator identifier, i.e., the same record regenerator.

[0582] Next, the save data storage and regeneration which perform an above-mentioned program use limit and a record regenerator use limit collectively are explained using drawing 80 -82.

[0583] Drawing 80 is a save data storage processing flow. In step S781, a contents identifier is read from contents data, a program use limit judging is performed in step S782, and a record regenerator limit judging is performed in step S783.

[0584] In the case of "those with a program use limit", and "with a record regenerator limit", in step S785, the save data encryption key Ksav is generated based on the both sides of a contents proper key (ex.Kcon) and a record regenerator proper key (Kdev). This can be obtained by Ksav=DES (MKx, Kcon XOR Kdev) etc. with the application of the master key MKx stored in the internal memory of Ksav= (Kcon XOR Kdev) or the

record regenerator 300. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0585] In step S786, "those with a program use limit", and in "having no record regenerator limit", a contents proper key (ex.Kcon) is used as the save data encryption key Ksav, or they generate the save data encryption key Ksav based on a contents proper key (ex.Kcon).

[0586] In the case of "with no program use limit" and "with a record regenerator limit", in step S787, a record regenerator proper key (Kdev) is used as the save data encryption key Ksav, or the save data encryption key Ksav is generated based on a record regenerator proper key (Kdev). Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0587] Furthermore, in step S787, "with no program use limit" and in "having no record regenerator limit", the system common key Ksys, for example, a system signature key, is used as the save data encryption key Ksav, or they generate the save data encryption key Ksav based on the system signature key Ksys. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0588] At step S789, with the save data encryption key Ksav generated by either of steps S785–S788, save data are enciphered and it is stored in a storage device.

[0589] Furthermore, at step S790, the limit information set up in steps S782 and S783 is stored in a data control file. A data control file serves as a configuration shown in drawing 81, and includes a data number, a contents identifier, a record regenerator identifier, a program use limit, and a record regenerator limit as an item.

[0590] Drawing 82 is a processing flow which reproduces the save data stored in the storage device by processing of drawing 80, and is performed. At step S791, the contents identifier of an executive program and a record regenerator identifier are read, and a contents identifier, a record regenerator identifier, a program use limit, and record regenerator limit information are read from the data control file shown in drawing 81 in step S792. In this case, by "it carries out", when a contents identifier is inharmonious, record regenerator limit information ends [a program use limit] processing by "it carries out", when a record regenerator identifier is inharmonious.

[0591] Next, at steps S793, S794, and S795, decode key generation processing is set as either of four modes of steps S796–S799 according to the record data of a data

control file.

[0592] In the case of "those with a program use limit", and "with a record regenerator limit", in step S796, the save data decryption key Ksav is generated based on the both sides of a contents proper key (ex.Kcon) and a record regenerator proper key (Kdev). Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav. In step S797, "those with a program use limit", and in "having no record regenerator limit", a contents proper key (ex.Kcon) is used as the save data decryption key Ksav, or they generate the save data decryption key Ksav based on a contents proper key (ex.Kcon). Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0593] In the case of "with no program use limit" and "with a record regenerator limit", in step S798, a record regenerator proper key (Kdev) is used as the save data decryption key Ksav, or the save data decryption key Ksav is generated based on a record regenerator proper key (Kdev). Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav. Furthermore, in step S799, "with no program use limit" and in "having no record regenerator limit", the system common key Ksys, for example, a system signature key, is used as the save data decryption key Ksav, or they generate the save data decryption key Ksav based on the system signature key Ksys. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0594] The decryption key generation algorithm used as that to which the processing algorithm corresponding to encryption key generation processing in these decryption key generation processings can decode the data which were applied and were enciphered based on the same contents proper key and the record regenerator proper key with the decode key generated based on the same contents proper key and the record regenerator proper key is applied.

[0595] At step S800, decode processing is performed using the save data decryption key generated in either of the above-mentioned steps S796-S799, and decode save data set to the record regenerator 300, and are reproduced and performed.

[0596] According to this drawing 80, the save data storage shown in 82, and the regeneration flow, it becomes that it is possible to decode the save data with which selection of "carrying out a program use limit" was made only when using contents

data with the same contents proper key, since encryption and decryption processing are performed with a contents proper key, and to use. Moreover, since encryption and decryption processing are performed by the record regenerator identifier, the save data with which selection of "carrying out a record regenerator limit" was made become possible [decoding and using with a record regenerator with the same record regenerator identifier, i.e. the same record regenerator,]. Therefore, it becomes possible to set up a use limit by contents and record regenerator both, and it becomes possible to raise the security of save data further.

[0597] In addition, in drawing 80 and 82, although the generation configuration of a contents proper key, the save data encryption key using record regenerator ******, and a decryption key was shown, it is good also as a configuration which uses a record regenerator identifier instead of a contents identifier and a record regenerator proper key instead of contents ******, and performs generation of a save data encryption key and a decryption key based on these identifiers.

[0598] Next, the configuration which generates a save data encryption key and a decryption key based on the password which the user entered using drawing 83 -85 is explained.

[0599] Drawing 83 is a processing flow which generates a save data encryption key based on the password which the user entered, and is stored in a storage device.

[0600] Step S821 is processing which reads a contents identifier from contents data, and is the same as that of each above-mentioned processing. Step S822 is a step which determines whether set up the program use limit by the user. The data control file set up in this configuration has the configuration shown in drawing 84.

[0601] As shown in drawing 84, as for data, a data number, a contents identifier, a record regenerator identifier, and the program use limit information according to a user further are included. "The program use limit information by the user" is an item which sets up whether the user who uses a program is restricted, or it does not carry out.

[0602] If a carrying-out-in step S822 in processing flow in drawing 83-use limit setup is made, the input of a user password will be made in step S823. This input is inputted from input means, such as a keyboard shown in drawing 2.

[0603] The entered password is outputted to the basis of control of Maine CPU 106 and a control section 301 at the cipher-processing section 302, and the save data encryption key Ksav based on the processing in step S824, i.e., an input user password, is generated. As save data encryption key Ksav generation processing, it is good also as an encryption key Ksav, or save data encryption key Ksav=DES (MKx,

password) may generate the password itself using the master key MKx of a record regenerator, for example. Moreover, on the other hand, a tropism function may be applied by considering a password as an input, and an encryption key may be generated based on the output.

[0604] When the user limit in step S822 is set to No, in step S828, a save data encryption key is generated based on the system common key of the record regenerator 300.

[0605] Furthermore, the save data with which save data encryption processing was made using the save data encryption key Ksav generated at step S824 or step S828, and encryption processing was made in step S826 at step S825 are stored in a storage device.

[0606] Furthermore, in step S827, the FUROGURAMU use limit information by the user who set it as the data control file of drawing 84 at step S822 is matched and written in a contents identifier and a record regenerator identifier.

[0607] Drawing 85 is drawing having shown the regeneration flow of the save data stored by processing of drawing 83. In step S831, a contents identifier is read from contents data and the program use limit information by the contents identifier and the user is read from the data control file shown in drawing 84 in step S832.

[0608] In step S833, when the judgment based on the data in a data control file is performed and "it is based on a user and a program use limit is carried out" is set up, in step S834, it asks for a password input and the decryption key based on an input password is generated in step S835. It is set as the decryption key generation algorithm used as that to which the processing algorithm corresponding to encryption key generation processing in this decryption key generation processing can decode the data which were applied and were enciphered based on a certain password with the decode key generated based on the same password.

[0609] Using the system common key Ksys stored in the internal memory of the record regenerator 300 in step S837, for example, a system signature key, when the judgment of step S833 has no program use limit by the user, the save data decode key Ksav is generated. Or the cryptographic key other than other keys saved in the internal memory 307 of the record regenerator 300 may be separately used as a save data cryptographic key Ksav.

[0610] At step S836, decode of the save data stored in the storage device using the decryption key Ksav generated in step S835 or step S837 is performed, and playback of save data and activation are made in a record regenerator in step S836.

[0611] Since encryption and decryption processing are performed with the key based

on a user input password, only when the save data with which selection of "it being based on a user and carrying out a program use limit" was made enter the same password according to the save data storage and the regeneration flow which were shown in drawing 83 and drawing 85, it becomes that it is possible in raising the security of ** and save data as it is possible in decoding and using.

[0612] As mentioned above, although storing processing of some save data and a regeneration mode have been explained, the mode which uses it for arbitration combining the processing which united the processing mentioned above, for example, a password, a record regenerator identifier, a contents identifier, etc., and generates a save data encryption key and a decryption key is also possible.

[0613] (17) exclusion (RIBOKESHON) **** of an inaccurate device -- as already explained, while the configuration which performs authentication, encryption processing, etc. and stores in a storage device various contents data offered from media 500 (refer to drawing 3) and means of communications 600 in the record regenerator 300 in the data processor of this invention raises the security of offer contents, only a just user has the configuration made available.

[0614] Authentication processing, encryption processing, and decryption processing are made using various signature keys stored in the internal memory 307 from which input contents are constituted by the cipher-processing section 302 of the record regenerator 300, a master key, and a check value generation key (refer to drawing 18) so that I may be understood from above-mentioned explanation. The internal memory 307 which stores this key information consists of semiconductor chips with the structure which is hard to access from the outside fundamentally, as explained previously. [whether the memory of the interior is pinched by dummy layers, such as an aluminum layer, by having multilayer structure, and] The width of face of the electrical potential difference or/which is constituted by the lowest layer and operates, and a frequency is narrow, Although it is desirable to consider read-out of data as the configuration made into the difficult property unjustly from the outside Unjust reading of an internal memory should be performed, these key data etc. flow out, and when copied to the record regenerator with which a regular license is not carried out, unjust contents use may be made using the copied key information.

[0615] Here, the configuration which prevents unjust use of the contents by the duplicate of the key by these illegal copies is explained.

[0616] The block diagram which explains this configuration "the exclusion configuration of (17) inaccurate devices" to drawing 86 is shown. The record regenerator 300 is the same as above-mentioned drawing 2 and the record

regenerator shown in 3, and it has an internal memory, and has the record regenerator identifier in various kinds of (drawing 18) key data and the pan which were explained previously. In addition, it does not restrict that the record regenerator identifier reproduced by the third person, key data, etc. are stored in the internal memory 307 shown in drawing 3 here, but the key data of the record regenerator 300 shown in drawing 86 etc. are gathered in the accessible memory section by the cipher-processing section 302 (drawing 2, 3 reference), or suppose that it is the configuration distributed and stored.

[0617] In order to realize the exclusion configuration of an inaccurate device, it considered as the configuration which memorized the unjust record regenerator identifier list of header units of contents data. As shown in drawing 86, to contents data, the RIBOKESHON (Revocation) list as an unjust record regenerator identifier (IDdev) list is held. Furthermore, the list check value ICVrev for the alteration check of a RIBOKESHON list is established. An unjust record regenerator identifier (IDdev) list list-izes the identifier IDdev of the inaccurate record regenerator with which the contents provider or the manager became clear from the circulation condition of an illegal copy etc. It is enciphered for example, with the delivery key Kdis, and this RIBOKESHON list may be stored. About the decode processing by the record regenerator, it is the same as that of the mode of contents download processing of previous drawing 22, for example.

[0618] In addition, although the RIBOKESHON list is shown in the contents data of drawing 86 as independent data in order to make an understanding easy, a RIBOKESHON list may be included here in the handling plan (for example, drawing 32 - 35 reference) which is the component of the header unit of the contents data explained previously, for example. In this case, the alteration check of the handling plan data which include a RIBOKESHON list with the check value ICVa explained previously is made. When a RIBOKESHON list is included in a handling plan, it is not necessary to be substituted by the check of check value A:ICVa, and for the check value A generation key Kicva in a record regenerator to be used, and to store check value generation key Kicv-rev.

[0619] When it includes a RIBOKESHON list in contents data as independent data, while performing the check of the RIBOKESHON list by the list check value ICVrev for the alteration check of a RIBOKESHON list, it carries out as the configuration which generates a middle check value from the list check value ICVrev and other partial check values in contents data, and performs verification processing of a middle check value.

[0620] The check technique of the RIBOKESHON list by the list check value ICVrev for the alteration check of a RIBOKESHON list can be performed by the same approach as check value generation processing of ICVa, ICVb, etc. in which it explained by above-mentioned drawing 23, drawing 24, etc. That is, it is calculated according to the ICV count approach which used as the key check value generation key Kicv-rev saved at the internal memory 307 of the record regenerator cipher-processing section 302, and was explained by drawing 23, drawing 24, etc. by making into a message the RIBOKESHON list included in contents data. Check value:ICV-rev stored in calculated check value ICV-rev' and a header (Header) is compared, and when in agreement, it judges with there being no alteration.

[0621] A middle check value including a list check value ICVrev generates with the application of the ICV count approach of having used as a key the total check value generation key Kicvt saved at the internal memory 307 of the record regenerator cipher-processing section 302, and having explained it to the check value A, the check value B, the list check value ICVrev, and the message train that applied the contents check value according to the format further in verified Header by drawing 7 etc., as shown in drawing 25.

[0622] The record regenerator 300 is provided with these RIBOKESHON lists and a list check value through the storage devices 400, such as a memory card, through the media 500, such as DVD and CD, and means of communications 600. The record regenerator 300 may have the case where it is the record regenerator which holds just key data, and the identifier ID reproduced unjustly here.

[0623] The processing flow of exclusion processing of the inaccurate record regenerator in such a configuration is shown in drawing 87 and drawing 88. Drawing 87 is an unjust record regenerator exclusion (RIBOKESHON) processing flow in case contents are offered from the media 500, such as DVD and CD, or means of communications 600, and drawing 88 is an unjust record regenerator exclusion (RIBOKESHON) processing flow in case contents are offered from the storage devices 400, such as a memory card.

[0624] First, the processing flow of drawing 87 is explained. Step 901 is a step which equips with media and performs the demand of offer of contents, i.e., regeneration, and download. Processing shown in this drawing 87 is performed as a step before equipping for example, a record regenerator with media, such as DVD, and performing download processing etc. About download processing, it is as explaining using drawing 22 previously, and processing of this drawing 87 is performed as the front step of activation of the processing flow of drawing 22, or processing inserted into the

processing flow of drawing 22.

[0625] When the record regenerator 300 receives contents offer through means of communications, such as a network, in step S911, the communication link session by the side of a contents distribution service is established, and it progresses to step S902 after that.

[0626] At step S902, a RIBOKESHON list (refer to drawing 86) is acquired from the header unit of contents data. The control section 301 shown in drawing 3 when contents are in media reads this list acquisition processing from media through a read station 304, and when contents are from means of communications, the control section 301 shown in drawing 3 receives from a contents distribution side through the communications department 305.

[0627] Next, a control section 301 makes the cipher-processing section 302 perform delivery and check value generation processing for the RIBOKESHON list acquired from media 500 or means of communications 600 in the cipher-processing section 302 in step S903. The record regenerator 300 has RIBOKESHON check value generation key Kicv-rev inside, and applies RIBOKESHON check value generation key Kicv-rev to it by making the RIBOKESHON list which received into a message. For example, check value ICV-rev' is calculated according to the ICV count approach explained by drawing 23, drawing 24, etc. Check value:ICV-rev stored in the header (Header) of a count result and contents data is compared, and when in agreement, it judges with there being no alteration (it being Yes at step S904). When not in agreement, it is judged with being altered and processing is ended as a progress transaction error to step S909.

[0628] Next, the control section 306 of the record regenerator cipher-processing section 302 makes the code / decryption section 308 of the record regenerator cipher-processing section 302 calculate total check value ICVt' in step S905. As shown in drawing 25, total check value ICVt' uses as a key the system signature key Ksys saved at the internal memory 307 of the record regenerator cipher-processing section 302, and enciphers and generates a middle check value by DES. In addition, although verification processing of each partial check value, for example, ICVa, ICVb, etc. is omitted in the processing flow shown in this drawing 87, verification of a partial check value according to each of the same data format as the processing flow of drawing 39 explained previously - drawing 45 is performed.

[0629] Next, in step S906, ICVt in generated total check value ICVt' and a header (Header) is compared, and when in agreement (it is Yes at step S906), it progresses to step S907. When not in agreement, it is judged with being altered and processing is

ended as a progress transaction error to step S909.

[0630] As explained previously, although the total check value ICVt checks ICVa, ICVb, and the whole partial check value included in contents data, such as a check value of each contents block, according to a data format, it verifies all an alteration of these for the list check value ICVrev further for the alteration check of a RIBOKESHON list as a partial check value to these partial check values further here. the case of being in agreement with check value:ICVt by which the total check value generated by above-mentioned processing was stored in the header (Header) -- ICVa, ICVb, the check value of each contents block, and the list check value ICVrev -- it is judged that all alterations cannot be found.

[0631] The comparison with the record regenerator identifier (IDdev) furthermore stored in the RIBOKESHON list judged that has no alteration, and the record regenerator 300 of self at step S907 is made.

[0632] When the identifier IDdev of the record regenerator of self is contained in the list of inaccurate record regenerator identifiers IDdev read from contents data, it judges that the record regenerator 300 has key data reproduced unjustly, it progresses to step S909, and future procedure is stopped. For example, activation of the procedure of contents download processing of drawing 22 is made impossible.

[0633] In step S907, when judged with the identifier IDdev of the record regenerator of self not being contained in the list of inaccurate record regenerator identifiers IDdev, it judges that the record regenerator 300 has just key data, it progresses to step S908, and activation of contents download processing of future procedure, for example, program execution processing, or drawing 22 etc. of it is attained.

[0634] Drawing 88 shows the processing in the case of reproducing the contents data stored in the storage devices 400, such as a memory card. As explained previously, mutual recognition processing (step S921) which explained a storage device 400 and the record regenerators 300, such as a memory card, by drawing 20 is performed. Step S922 It sets, only when it is mutual recognition O.K., it progresses to the processing after step S923, and when mutual recognition goes wrong, it becomes the error of step S930 and subsequent processings are not performed.

[0635] At step S923, a RIBOKESHON list (refer to drawing 86) is acquired from the header unit of contents data. Processings of the subsequent steps S924-S930 are the correspondence processing in previous drawing 87, and the same processing. Namely, verification of the list by the list check value (S924, S925), The comparison (S928) with the entry of the verification (S926, S927) by the total check value and a list and the record regenerator identifier IDdev of self is performed. When the

identifier IDdev of the record regenerator of self is contained in the list of inaccurate record regenerator identifiers IDdev read from contents data It judges that the record regenerator 300 has key data reproduced unjustly, it progresses to step S930, and future procedure is stopped. For example, activation of regeneration of the contents shown in drawing 28 is made impossible. When judged with on the other hand the identifier IDdev of the record regenerator of self not being contained in the list of inaccurate record regenerator identifiers IDdev, it judges that the record regenerator 300 has just key data, it progresses to step S929, and activation of future procedure is attained.

[0636] Thus, it sets to the data processor of this invention. The data which combine with the contents which a contents provider or a manager offers, and identify an inaccurate record regenerator, Namely, include the RIBOKESHON list which list-ized the inaccurate record regenerator identifier IDdev as configuration data of the header unit of contents data, and it provides for a record regenerator user. The record regenerator identifier IDdev by which the record regenerator user was stored in the memory of the record regenerator of self in advance of use of the contents by the record regenerator When the data which perform collating with the identifier of a list and are in agreement exist, since it considered as the configuration which does not perform future processings, it becomes possible to eliminate the contents use by the inaccurate record regenerator which reproduced key data and was stored in memory.

[0637] (18) As explained to a secure chip configuration and the manufacture approach point, since the internal memory 307 of the record regenerator cipher-processing section 302 or the internal memory 405 of a storage device 400 holds important information, such as a cryptographic key, it is necessary to make it into the structure which is hard to read unjustly from the exterior. Therefore, the record regenerator cipher-processing section 302 and the storage device cipher-processing section 401 For example, it consists of semiconductor chips with the structure which is hard to access from the outside. It has multilayer structure and read-out of data is constituted as Tampa-proof memory which has a difficult property unjustly [that the memory of the interior has the narrow width of face of the electrical potential difference or/which is inserted into dummy layers, such as an aluminum layer, or is constituted by the lowest layer and operates, and a frequency etc.] from the outside.

[0638] However, it is necessary for the internal memory 307 of the record regenerator cipher-processing section 302 to write in different data for every record regenerators, such as the record regenerator signature key Kdev, so that I may be understood by above-mentioned explanation. Moreover, after writing the individual information for

every chip, for example, identification information (ID) and cryptographic key information, in the storage region of the non-volatile in a chip, for example, a flash memory, FeRAM, etc., it is necessary to make difficult re-writing of the data after product shipment, and read-out.

[0639] The command protocol of for example, data writing is made secret at the technique for making difficult read-out of the conventional write-in data and re-write-in processing. Or the signal line which receives the data write command on a chip, and the signal line for a communication link used after producing commercially are separated and constituted, and unless a direct signal is sent to the chip on a substrate, there is the technique of making it a data write command not become effective etc.

[0640] However, even if it adopts such conventional technique, whenever there are facility and the technique of making a circuit driving, for what has the know how of a storage element, even if the signal output to the data write-in field of a chip is possible and the command protocol of data writing is secret, the analysis possibility of a protocol will exist.

[0641] Circulating the storing component of the code processed data holding such alteration possibility of restricted data results in threatening the whole code processing system. Moreover, although it is also possible to consider as the configuration which does not mount the data read-out command itself in order to prevent read-out of data in that case, even if it is the case where the data writing of normal is performed, check whether the data writing to memory has actually been performed, or It becomes impossible to judge whether the written-in data are written in correctly, and possibility that the chip with which poor data writing was performed will be supplied occurs.

[0642] While enabling exact data writing in view of these conventional techniques here at nonvolatile memory, such as a flash memory and FeRAM, the secure chip configuration and the secure chip manufacture approach of making read-out of data difficult are offered.

[0643] A security chip configuration applicable to drawing 89 at the above-mentioned record regenerator cipher-processing section 302 or the cipher-processing section 401 of a storage device 400 is shown. Drawing 89 (A) shows the security chip configuration in the manufacture process of a chip, i.e., the write-in process of data, and drawing 89 (B) shows the example of the example 300 of a configuration of the product carrying the security chip which wrote in data, for example, a record regenerator, and a storage device 400.

[0644] As for the security chip in a manufacture process, the signal line 8003 for mode assignment and the various command signal lines 8004 are connected at the processing section 8001, and the processing section 8001 performs data write-in processing to the storage section 8002 which is nonvolatile memory, or data read-out processing from the storage section 8002 according to the mode set up with the signal line 8003 for mode assignment, for example, a data write mode, and data read-out mode.

[0645] On the other hand, in the security chip loading product of drawing 89 (B), although an external connection interface, a peripheral device, other components, etc. are connected with a security chip with a general-purpose signal line, the mode signal line 8003 is made into a connectionless condition. Concrete processing is cutting or sealing the signal line which carries out grounding of the mode signal line 8003 and which has been fished to Vcc by insulator resin etc. By such processing, access of as opposed to the mode signal line of a security chip in after product shipment becomes difficult, and can raise the difficulty of reading the data of a chip from the exterior or writing in.

[0646] Furthermore, the security chip 8000 of this configuration has the configuration which makes difficult write-in processing to the data storage section 8002, and read-out processing of the data written in the storage section 8002, and even if it is the case where a third person succeeds in access of the mode signal line 8003 even if, it can prevent unjust data writing and read-out. The data writing or read-out processing flow in the security chip which has this configuration in drawing 90 is shown.

[0647] Step S951 is a step which sets the mode signal line 8003 as a data write mode or data read-out mode.

[0648] Step S952 is a step which takes out the information for authentication from a chip. Information required for authentication processing, such as key information for authentication processing in a password and a code technique, is beforehand stored in the security chip of this configuration by the wire (Wire) and the mask-ROM configuration. Step S952 reads this authentication information, and performs authentication processing. For example, when a regular data write-in fixture and data read-out equipment are connected to a general-purpose signal line and authentication processing is performed, the result of Authentication O.K. (it sets to step S953 and is Yes) is obtained, but when an inaccurate data write-in fixture and data read-out equipment are connected to a general-purpose signal line and authentication processing is performed, it fails to authentication (it sets to step S953

and is No), and processing is stopped at the time. Authentication processing can be performed according to the mutual recognition processing procedure of drawing 13 explained previously, for example. The processing section 8001 shown in drawing 89 has the configuration which can perform these authentication processings. This is realizable with the same configuration as the command register built into the control section 403 of the cipher-processing section 401 of the storage device 400 shown in drawing 29 explained previously. For example, the processing section of the chip of drawing 89 has the same configuration as the command register built into the control section 403 of the cipher-processing section 401 of the storage device 400 shown in drawing 29, and becomes possible [performing corresponding processing and performing an authentication processing sequence], if the predetermined command No is inputted from the device connected to the various command signal lines 8004.

[0649] Only when authentication is made in authentication processing, the processing section 8001 receives the write command of data, or the read-out command of data, and performs write-in processing (step S955) of data or read-out processing (step S956) of data.

[0650] Thus, in the security chip of this configuration, since it considered as the configuration which performs authentication processing at the time of read-out at the time of the writing of data, read-out of data or the data writing to the storage section can be prevented from the storage section of the security chip by the third person without a just right.

[0651] Next, the example considered as the component configuration with still higher security is shown in drawing 91. In this example, the storage section 8200 of a security chip is divided into two fields, one side is the read-out write-in concomitant use field (RW:ReadWrite field) 8201 which can write data, and another side is the field 8202 only for writing (WO:WriteOnly field) which can only write in data.

[0652] In this configuration, high data of a security request, such as cryptographic key data and identifier data, are written in the field 8202 only for writing (WO:WriteOnly field), and on the other hand, the data for a check of whenever [security] etc. are read and it writes in the write-in concomitant use field (RW:ReadWrite field) 8201 so highly.

[0653] The processing section 8001 performs data read-out processing accompanied by the authentication processing which explained the data read-out processing from the read-out write-in concomitant use field (RW:ReadWrite field) 8201 by above-mentioned drawing 90. However, data write-in processing is performed according to the flow of drawing 92.

[0654] Step S961 of drawing 92 is a step which sets the mode signal line 8003 as a write mode, and performs same authentication processing with previous drawing 90 having explained at step 962. If authentication is made by authentication processing, it progresses to step S963, and through the command signal line 8004, whenever [security] will not be so high to the writing of the information on the high key data of security etc., and the read-out write-in concomitant use field (RW:ReadWrite field) 8201, for example, the data writing **** command for a check will be outputted to the field 8202 only for writing (WO) to the processing section 8001 to them.

[0655] At step S964, the processing section 8001 which received the command performs data write-in processing according to a command to the field 8202 only for writing (WO), and the read-out write-in concomitant use field (RW:ReadWrite field) 8201, respectively.

[0656] Moreover, the verification processing flow of the data written in the field 8202 only for writing (WO) is shown in drawing 93.

[0657] Step S971 of drawing 93 performs cipher processing based on the data written in the field 8202 only for writing (WO) in the processing section 8001. These activation configurations are realized by the configuration which carries out sequential execution of the cipher-processing sequence stored in the command register as well as a previous authentication processing activation configuration. Moreover, especially the cipher-processing algorithm performed in the processing section 8001 is not limited, and can be considered as the configuration which performs the DES algorithm explained previously.

[0658] Next, the verification equipment connected to the security chip receives a cipher-processing result from the processing section 8001 at step S972. Next, in step S973, the result obtained with the application of the same encryption processing as the algorithm performed in the processing section 8001 to the regular write-in data which performed write-in processing in the storage section previously is compared with the encryption result from the processing section 8001.

[0659] If the compared result is the same, it will be verified that the data written in the field 8202 only for writing (WO) are right.

[0660] With this configuration, even if authentication processing should be broken, it should read and activation of a command should be attained, the field of data which can be read is limited to the read-out write-in concomitant use field (RW:ReadWrite field) 8201, and read-out of the data written in the field 8202 only for writing (WO) is impossible, and serves as a high configuration of security further. Moreover, since the read-out write-in concomitant use field (RW:ReadWrite field) 8201 is constituted

unlike the chip which completely made read-out impossible, the right-or-wrong check of memory access is possible.

[0661] As mentioned above, it has explained in detail about this invention, referring to a specific example. However, it is obvious that this contractor can accomplish correction and substitution of this example in the range which does not deviate from the summary of this invention. That is, with the gestalt of instantiation, this invention has been indicated and it should not be interpreted restrictively. Moreover, although the possible record regenerator was made into the example and the above-mentioned example has explained record of contents, and playback, as for the configuration of this invention, only data logging can apply only data playback also in possible equipment, and this invention can be carried out in the various general data processors of a personal computer, a game device, and others. In order to judge the summary of this invention, the column of the claim indicated at the beginning should be taken into consideration.

[0662]

[Effect of the Invention] Thus, according to the data processor of this invention, the data-processing approach, and the contents data verification value grant approach Per contents block data, generate a contents check value and collating processing of the generated contents check value is performed. Furthermore, since it considered as the configuration which generates a contents check value by cipher processing which generated the contents mean value based on the contents block data for verification, and applied the contents check value generation key, efficient verification is attained as compared with processing of the conventional whole data.

[0663] Furthermore, according to the data processor of this invention, the data-processing approach, and the contents data verification value grant approach, while becoming verifiable in a contents block unit, the use mode of contents data, for example, download processing, and the simplified verification processing according to regeneration are attained, and efficient verification which **(ed) in the use mode can be performed.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is drawing showing the configuration of the conventional data processing system.

[Drawing 2] It is drawing showing the configuration of the data processor with which this invention is applied.

[Drawing 3] It is drawing showing the configuration of the data processor with which this invention is applied.

[Drawing 4] It is drawing showing the data format of the contents data on media and a channel.

[Drawing 5] It is drawing showing the handling plan contained in the header in contents data.

[Drawing 6] It is drawing showing the block information included in the header in contents data.

[Drawing 7] It is drawing showing the electronic signature generation method using DES.

[Drawing 8] It is drawing showing the electronic signature generation method using Triple DES.

[Drawing 9] It is drawing explaining the mode of Triple DES.

[Drawing 10] It is drawing showing the electronic signature generation method which used Triple DES for the part.

[Drawing 11] It is drawing showing the processing flow in electronic signature generation.

[Drawing 12] It is drawing showing the processing flow in electronic signature verification.

[Drawing 13] It is drawing explaining the processing sequence of the mutual recognition processing using a symmetry key code technique.

[Drawing 14] It is drawing explaining a public key certificate.

[Drawing 15] It is drawing explaining the processing sequence of the mutual recognition processing using an unsymmetrical key code technique.

[Drawing 16] It is drawing showing the processing flow of the encryption processing using an elliptic curve cryptosystem.

[Drawing 17] It is drawing showing the processing flow of the decryption processing using an elliptic curve cryptosystem.

[Drawing 18] It is drawing showing the data-hold situation on a record regenerator.

[Drawing 19] It is drawing showing the data-hold situation on a storage device.

[Drawing 20] It is drawing showing the mutual recognition processing flow of a record regenerator and a storage device.

[Drawing 21] It is drawing showing the relation between the master key of a record regenerator, and the correspondence key block of a storage device.

[Drawing 22] It is drawing showing the processing flow in download processing of contents.

[Drawing 23] It is drawing explaining the generation method of check value A:ICVa.

[Drawing 24] It is drawing explaining the generation method of check value B:ICVb.

[Drawing 25] It is drawing explaining the generation method of the total check value and a record regenerator proper check value.

[Drawing 26] It is drawing showing a format (use limit information = 0) of the contents data saved at the storage device.

[Drawing 27] It is drawing showing a format (use limit information = 1) of the contents data saved at the storage device.

[Drawing 28] It is drawing showing the processing flow in regeneration of contents.

[Drawing 29] It is drawing explaining the command execution approach in a storage device.

[Drawing 30] It is drawing explaining the command execution approach in the contents storing processing in a storage device.

[Drawing 31] It is drawing explaining the command execution approach in the contents regeneration in a storage device.

[Drawing 32] It is drawing explaining the format type 0 configuration of a contents data format.

[Drawing 33] It is drawing explaining the format type 1 configuration of a contents data format.

[Drawing 34] It is drawing explaining the format type 2 configuration of a contents data format.

[Drawing 35] It is drawing explaining the format type 3 configuration of a contents data format.

[Drawing 36] It is drawing explaining the generation art of the contents check value

ICVi in the format type 0.

[Drawing 37] It is drawing explaining the generation art of the contents check value ICVi in the format type 1.

[Drawing 38] It is drawing explaining the generation art of the total check value in the format types 2 and 3, and a record regenerator proper check value.

[Drawing 39] It is drawing showing the processing flow of the contents download processing in the format types 0 and 1.

[Drawing 40] It is drawing showing the processing flow of the contents download processing in the format type 2.

[Drawing 41] It is drawing showing the processing flow of the contents download processing in the format type 3.

[Drawing 42] It is drawing showing the processing flow of the contents regeneration in the format type 0.

[Drawing 43] It is drawing showing the processing flow of the contents regeneration in the format type 1.

[Drawing 44] It is drawing showing the processing flow of the contents regeneration in the format type 2.

[Drawing 45] It is drawing showing the processing flow of the contents regeneration in the format type 3.

[Drawing 46] It is drawing (the 1) explaining generation of the check value in a contents generation person and a contents verification person, and the verification approach.

[Drawing 47] It is drawing (the 2) explaining generation of the check value in a contents generation person and a contents verification person, and the verification approach.

[Drawing 48] It is drawing (the 3) explaining generation of the check value in a contents generation person and a contents verification person, and the verification approach.

[Drawing 49] It is drawing explaining how to generate various kinds of keys according to an individual using a master key.

[Drawing 50] It is drawing (Example 1) showing the example of processing in a user with a content provider about how to generate various kinds of keys according to an individual using a master key.

[Drawing 51] It is drawing (Example 2) showing the example of processing in a user with a content provider about how to generate various kinds of keys according to an individual using a master key.

[Drawing 52] It is drawing which explains the configuration which performs a use limit by proper use of a master key.

[Drawing 53] It is drawing (Example 3) showing the example of processing in a user with a content provider about how to generate various kinds of keys according to an individual using a master key.

[Drawing 54] It is drawing (Example 4) showing the example of processing in a user with a content provider about how to generate various kinds of keys according to an individual using a master key.

[Drawing 55] It is drawing (Example 5) showing the example of processing in a user with a content provider about how to generate various kinds of keys according to an individual using a master key.

[Drawing 56] It is drawing showing the processing flow which stores the cryptographic key which applied Triple DES using a single DES algorithm.

[Drawing 57] It is drawing showing the contents regeneration flow (Example 1) based on priority.

[Drawing 58] It is drawing showing the contents regeneration flow (Example 2) based on priority.

[Drawing 59] It is drawing showing the contents regeneration flow (Example 3) based on priority.

[Drawing 60] It is drawing explaining the configuration which performs decode (expanding) processing of the compressed data in contents regeneration.

[Drawing 61] It is drawing showing the example of a configuration of contents (Example 1).

[Drawing 62] It is drawing showing the regeneration flow in the example 1 of a configuration of contents.

[Drawing 63] It is drawing showing the example of a configuration of contents (Example 2).

[Drawing 64] It is drawing showing the regeneration flow in the example 2 of a configuration of contents.

[Drawing 65] It is drawing showing the example of a configuration of contents (Example 3).

[Drawing 66] It is drawing showing the regeneration flow in the example 3 of a configuration of contents.

[Drawing 67] It is drawing showing the example of a configuration of contents (Example 4).

[Drawing 68] It is drawing showing the regeneration flow in the example 4 of a

configuration of contents.

[Drawing 69] It is drawing explaining generation of save data, and storing processing.

[Drawing 70] It is drawing showing the processing flow about the example of storing processing of save data (Example 1).

[Drawing 71] It is drawing showing the data control file organization (Example 1) used in storing of save data, and regeneration.

[Drawing 72] It is drawing showing the processing flow about the example of regeneration of save data (Example 1).

[Drawing 73] It is drawing showing the processing flow about the example of storing processing of save data (Example 2).

[Drawing 74] It is drawing showing the processing flow about the example of regeneration of save data (Example 2).

[Drawing 75] It is drawing showing the processing flow about the example of storing processing of save data (Example 3).

[Drawing 76] It is drawing showing the data control file organization (Example 2) used in storing of save data, and regeneration.

[Drawing 77] It is drawing showing the processing flow about the example of regeneration of save data (Example 3).

[Drawing 78] It is drawing showing the processing flow about the example of storing processing of save data (Example 4).

[Drawing 79] It is drawing showing the processing flow about the example of regeneration of save data (Example 4).

[Drawing 80] It is drawing showing the processing flow about the example of storing processing of save data (Example 5).

[Drawing 81] It is drawing showing the data control file organization (Example 3) used in storing of save data, and regeneration.

[Drawing 82] It is drawing showing the processing flow about the example of regeneration of save data (Example 5).

[Drawing 83] It is drawing showing the processing flow about the example of storing processing of save data (Example 6).

[Drawing 84] It is drawing showing the data control file organization (Example 4) used in storing of save data, and regeneration.

[Drawing 85] It is drawing showing the processing flow about the example of regeneration of save data (Example 6).

[Drawing 86] It is drawing explaining a contents inaccurate user exclusion (RIBOKESHON) configuration.

[Drawing 87] It is drawing showing the processing flow (Example 1) of contents inaccurate user exclusion (RIBOKESHON).

[Drawing 88] It is drawing showing the processing flow (Example 2) of contents inaccurate user exclusion (RIBOKESHON).

[Drawing 89] It is drawing explaining the configuration (Example 1) of a security chip.

[Drawing 90] It is drawing showing the processing flow in the manufacture approach of a security chip.

[Drawing 91] It is drawing explaining the configuration (Example 2) of a security chip.

[Drawing 92] It is drawing showing the processing flow in the data write-in processing in a security chip (Example 2).

[Drawing 93] It is drawing showing the processing flow in the write-in data check processing in a security chip (Example 2).

[Description of Notations]

106 Maine CPU
107 RAM
108 ROM
109 AV Processing Section
110 Input-Process Section
111 PIO
112 SIO
300 Record Regenerator
301 Control Section
302 Cipher-Processing Section
303 Record Device Controller
304 Reading Section
305 Communications Department
306 Control Section
307 Internal Memory
308 Code / Decryption Section
400 Storage Device
401 Cipher-Processing Section
402 External Memory
403 Control Section
404 Communications Department
405 Internal Memory
406 Code / Decryption Section

407 External Memory Control Section
500 Media
600 Means of Communications
2101, 2102, 2103 Record regenerator
2104, 2105, 2106 Storage device
2901 Command Number Management Department
2902 Command Register
2903 2904 Authentication flag
3001 Loudspeaker
3002 Monitor
3090 Memory
3091 Contents Analysis Section
3092 Data Storage Section
3093 Program Store Section
3094 Compression Expanding Processing Section
7701 Contents Data
7702 RIBOKESHON List
7703 List Check Value
8000 Security Chip
8001 Processing Section
8002 Storage Section
8003 Mode Signal Line
8004 Command Signal Line
8201 Read-out Write-in Concomitant Use Field
8202 Field Only for Writing